DLD

Lecture No: 01

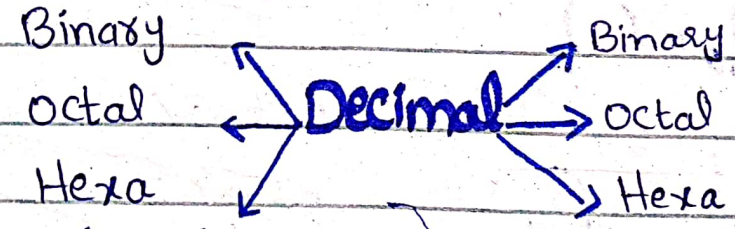
Number System

↳ Counting and representation of quantity.

↳ Starts from unary system and extends to decimal system.

Table	Radix/Base	Range
Binary	2	0, 1
Octal	8	0 → 7
Decimal	10	0 → 9
Hexa decimal	16	0 → 9
		10 → A
		11 → B
		12 → C
		13 → D

Conversions:



Binary to decimal:

1- $(1101)_2$

$2^3 \times 1 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$

$8 + 4 + 0 + 1$

$(13)_{10}$

2- $(11001.11)_2$

$2^4 \times 1 + 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 1 . 1 \times 2^{-1} + 1 \times 2^{-2}$

$16 + 8 + 0 + 0 + 1 . \frac{1}{2} + \frac{1}{4}$

$25 . 0.5 + 0.25 = (25.75)_{10}$

Octal to Decimal:

1- $(101.1)_8$

$8^2 \times 1 + 8^1 \times 0 + 8^0 \times 1 . 1 \times 8^{-1}$

$64 + 0 + 1 . \frac{1}{8}$

$(65.125)_{10}$

Hexadecimal to decimal:

1- $(10A)_{16}$

$16^2 \times 1 + 16^1 \times 0 + 16^0 \times A$ $\quad \because A = 10$

$256 + 0 + 10$

$(266)_{10}$

Decimal to Binary:

1- $(19)_{10} = (\ )_2$

2	19	
2	9	1
2	4	1
2	2	0
	1	0

$(10011)_2$

2- $(27.625)_{10} = (\ )_2$

2	27	
2	13	1
2	6	1
2	3	0
	1	1

$(11011.10)_2$

$0.625 \times 2 = 1.250$

$1.250 \times 2 = 0.50$

Decimal into hexadecimal:

1- $(1247.78)_{10} = (\ )_{16}$

Firstly:

$(4DE.C7A)_{16}$

16	1247
16	77 — 15 (E)
	4 — 13 (D)

$.78 \times 16 = 12.48$

$.48 \times 16 = 7.68$

$.68 \times 16 = 10.88$

Decimal to Octal:

$(1011.27)_{10} = (\ )_8$

8	1011
8	126 — 3
8	15 — 6
	1 — 7

$(1763.212)_8$

$.27 \times 8 = 2.16$

$.16 \times 8 = 1.28$

$.28 \times 8 = 2.24$

Lecture No: 02

Conversions:

Octal to Binary

Table:

0 ——— 000

1 ——— 001

2 ——— 010

3 ——— 011

4 ——— 100

5 ——— 101

6 ——— 110

7 ——— 111

1- $(735)_8 = (111011101)_2$

Binary to Octal → $(010110110001)_2 = (2661)_8$

Table: Hexadecimal to Binary

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

1- $(7BC)_{16}$

$= (011110111100)_2$

Binary to hexadecimal:

1- $(0010011101011100)_2$

$(23AC)_2$

Addition:

1- $786_{(10)} + 687_{(10)} = (1473)_{10}$

2- $741_{(8)} + 456_{(8)} = (1417)_8$

3- $11001_{(2)} + 10111_{(2)} = (110000)_2$

Subtraction

1- $786_{(10)}$
$-687_{(10)}$
$(99)_{10}$

2- $1000010_{(2)}$
$-010111_{(2)}$
$(001011)_2$

3- $613_{(8)}$
$-245_{(8)}$
$(346)_8$

4- $F43B_{(16)}$
$-5A7D_{(16)}$
$9BE$

05 Jan, 2024 Lecture No: 03

Signed number Representation

1- Signed magnitude
2- 1's complement
3- 2's complement

Table For Positive

Decimal #	Signed magnitude	1's Complement	2's Complement
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100

Decimal #	Signed magnitude	1's Complement	2's Complement
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000

Table for negative

Decimal #	Signed magnitude	1's Complement	2's Complement
-0	1000	1111	0000
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001

Remember that:

+ → 0 (start from 0 then sign is +)

- → 1 (start from 1 the sign is -)

Example:

Convert in Sign magnitude

1- 111

(1 = Sign bit = −)

−3

2- 1011

−3

06th Jan, 2024 Lecture No: 04

1's Complement:

−5 ⟶ 1's Complement

1010 ⟶ −5

0101 ⟶ +5

+69

01000101

−37 1011010

−23 101000

−22 10111

−27 100100

−43 1010100

+8 01000

2	69
2	34 − 1
2	12 − 1
2	6 − 0
2	3 − 0
	1 − 1

8 characters:

−5 = 11111010 Sign 1st Complement

+5 = 00000101 Bit of original Binary

2's Complement:

-5

+5

0101

1010 → 1's Complement

+1

1011 → 2's Complement

Reverse process:

1011 → 2's complement

-1

1010

↓

+5

Signed magnitude	1's Complement	2's Complement
-5	1010	1011
+69	01000101	01000101
-37	1011010	1011011
-23	101000	101001
-8	10111	11000
-27	100100	100101
-1/3	1010100	1010101
+8	01000	01000

12 Jan, 2024 Lecture No:05

Logic gates:

→ A logic gate is a device that acts as a building block for digital circuit.

→ They perform basic logical functions:

1- AND Gate:

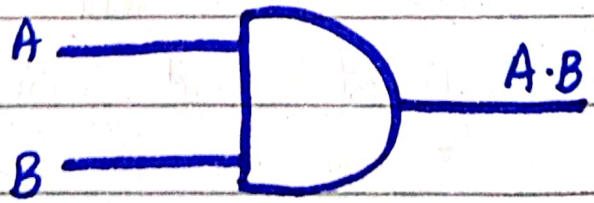
→ This gate works in the same way as logical operator "And."

→ It has minimum of two input values and Output Value

- **Truth table:**

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

- **Diagram:**



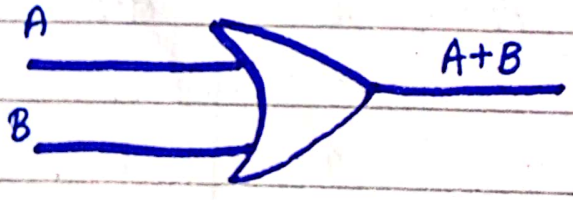
2- OR Gate:

→ This gate works on the same way as logical operator "OR".
→ It has minimum of two input values and output value.

- **Truth Table:**

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

- **Diagram:**



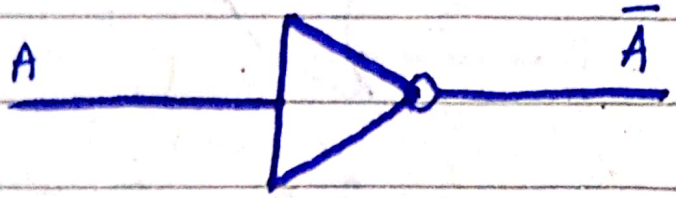
3- NOT Gate:

→ This gate is also called Investor
→ This gate give the inverse value of input value as a result.
→ It has one input and one output values.

- **Truth Table:**

A	$\bar{A}$
0	1
1	0

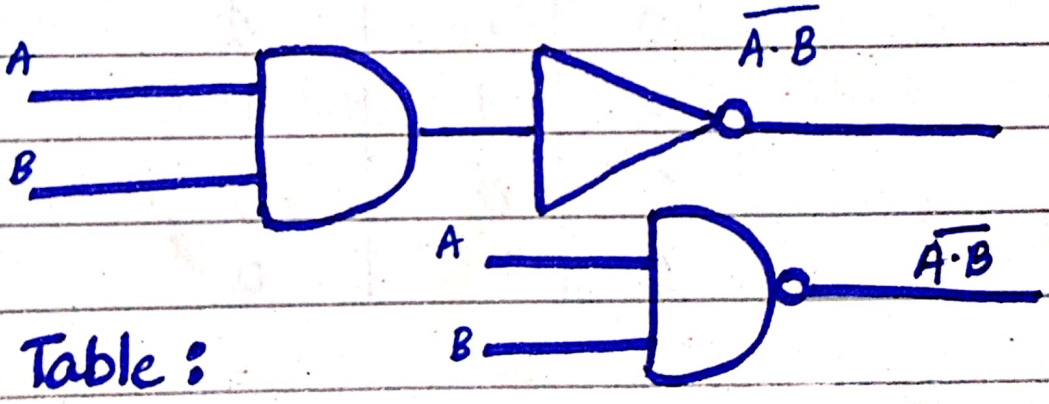
- Diagram:



4- NAND Gate:

It is a combination of "AND" and "NOT" gate.

- Diagram:



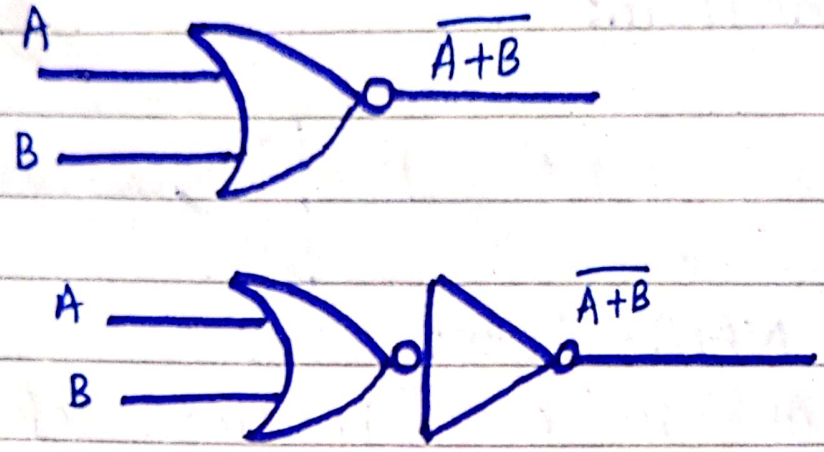
- Truth Table:

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

5- NOR Gate:

→ It is a combination of "OR" and "NOT" gate.

• **Diagram:**



• **Truth Table:**

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

6- XOR Gate:

→ It is also called Exclusive OR gate and Odd 1 Detector.

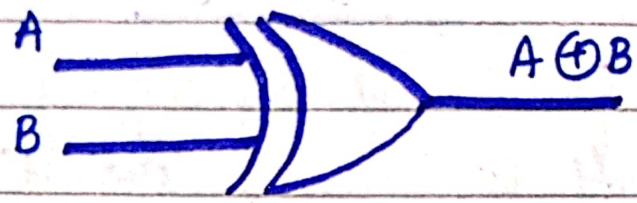
• **Truth Table:**

on next Page →

A	B	$A \oplus B = \bar{A}B + A\bar{B}$
0	0	0
0	1	1
1	0	1
1	1	0

جہاں 1 تعداد odd ہوگی تو output 1 ہوگی۔

- **Diagram:**



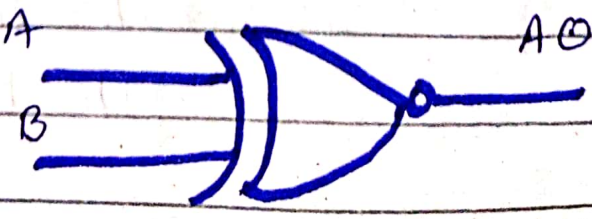
7- XNOR Gate:

It is also called exclusive NOR gate and even 1 detector

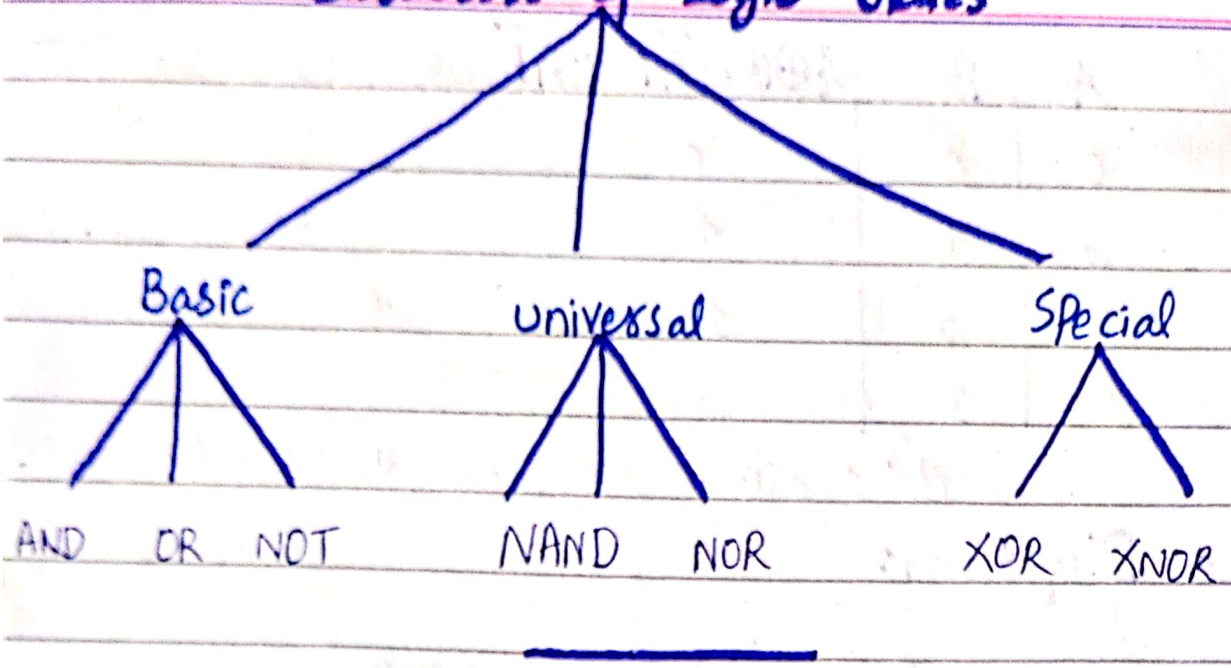
- **Truth Table:**

A	B	$A \odot B = \bar{A}\bar{B} + AB$
0	0	1
0	1	0
1	0	0
1	1	1

- **Diagram:**



Structure of Logic Gates



12 Jan, 2024 Lecture No: 06

1- Boolean Algebra:

It is a branch of mathamatics that deals with operations or on logical values with binary variables.

2- Boolean laws:

⇒ Idempotent Law:

$a \cdot a = a$

$a + a = a$

⇒ Identity Law:

$a + 0 = a$

$a \cdot 0 = 0$

$a + 1 = 1$

$a \cdot 1 = a$

⇒ Associative Law:

$a \cdot (b \cdot c) = (a \cdot b) \cdot c$

$a + (b + c) = (a + b) + c$

⇒ Commutative Law:

$a \cdot b = b \cdot a$

$a + b = b + a$

⟹ Distributive Law:

$a \cdot (b+c) = (a \cdot b) + (a \cdot c)$

$a + (b \cdot c) = (a+b) \cdot (a+c)$

⟹ Demorgan's Law:

$\overline{a+b} = \bar{a} \cdot \bar{b}$

$\overline{a \cdot b} = \bar{a} + \bar{b}$

⟹ complement Law:

$\bar{0} = 1$

$\bar{1} = 0$

$a \cdot \bar{a} = 0$

$a + \bar{a} = 1$

⟹ Involution Law:

$(\bar{\bar{a}}) = a$

---

18 Jan, 2024 Lecture No: 07

Boolean Expression:

It is an expression that evaluate a condition and result in a boolean value. that is true or False.

Method # 01

- Sum of Product (SOP)
- Also called disjunctive normal form
- Expression contains product (AND) terms which are sum (OR) together e.g $AB + A\bar{B}C + BC$
- Each product term consists of one or more literals appearing either in complemented or uncomplemented form.
- The product term which contain all

literals of function is called minterm.

A	1
$\bar{A}$	0

Minterm table

0	000	$\bar{A}\bar{B}\bar{C}$	m_0
1	001	$\bar{A}\bar{B}C$	m_1
2	010	$\bar{A}B\bar{C}$	m_2
3	011	$\bar{A}BC$	m_3
4	100	$A\bar{B}\bar{C}$	m_4
5	101	$A\bar{B}C$	m_5
6	110	$AB\bar{C}$	m_6
7	111	ABC	m_7

Canonical SOP (Standard SOP)

- In SOP expression, if each product term is also a minterm, Then it is called Canonical or Standard SOP.

Next table →

ABC	f
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	0

$\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

↓

minterm

- $A\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

$101 + 100 \quad 001$

$5 \quad 4 \quad 1$

$f(A,B,C) = \Sigma(1,4,5)$

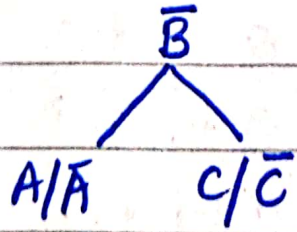
$f(A,B,C) = AB + A\bar{B}C$ (SOP)

↓ ↓

$ABC + AB\bar{C} + A\bar{B}C$ (SSOP)

- $f(A,B,C) = \bar{B} + A\bar{C}$ (SOP)

$A\bar{B}C + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + AB\bar{C} + A\bar{B}\bar{C}$



$A\bar{B}C, \bar{A}\bar{B}C, A\bar{B}\bar{C}, \bar{A}\bar{B}\bar{C}$

20 Jan, 2024 Lecture No:08

Product of Sum (POS)

→ Also called Conjunctive Normal form. Contain Sum (OR) terms, which are product (AND) together e.g. $(A+B)(B+\bar{C})$

→ Each Sum term consists of one or more literals either in complement or uncomplement form.

→ Sum terms which contains all literals is called Maxterm.

$a=0$; $\bar{a}=1$

Maxterm Table

0	000	$a+b+c$	m_0
1	001	$a+b+\bar{c}$	m_1
2	010	$a+b+c$	m_2
3	011	$a+\bar{b}+\bar{c}$	m_3
4	100	$a+b+c$	m_4
5	101	$\bar{a}+b+c$	m_5
6	110	$\bar{a}+\bar{b}+\bar{c}$	m_6
7	111	$\bar{a}+\bar{b}+\bar{c}$	m_7

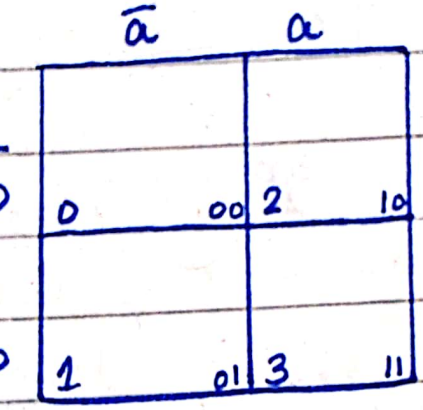
25 Jan, 2024 Lecture No: 09

K-map (Knaugh map)

→ Graphical representation of truth table
→ For n variable K-map there will 2^n cells addressed by gray code.

2 Variable K-map:

جس پر (-) یا بتوں کا (0) ہو



∵ $\bar{a}=0$, $\bar{b}=1$

3 variable K-map:

	$\bar{b}\bar{c}_{00}$	$\bar{b}c_{01}$	bc_{11}	$b\bar{c}_{10}$
$\bar{a}$ 0	0 (000)	1 (001)	3 (011)	2 (010)
a 1	4 (100)	5 (101)	7 (111)	6 (110)

OR

bc	$\bar{a}$	a
$\bar{b}\bar{c}$ 00	0 000	4 100
$\bar{b}c$ 01	1 001	5 101
bc 11	3 011	7 111
$b\bar{c}$ 10	2 010	6 110

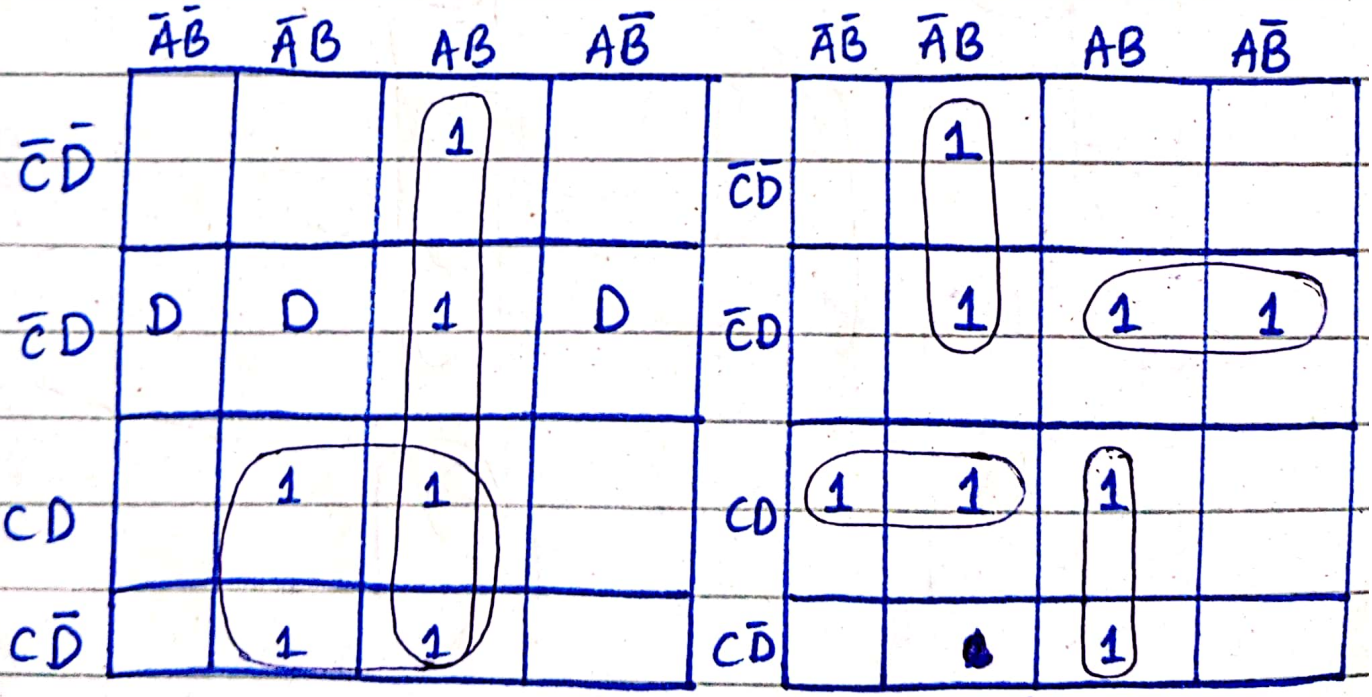
c \ ab	$\bar{a}\bar{b}$ 00	$\bar{a}b$ 01	ab 11	$a\bar{b}$ 10
$\bar{C}_0$	0 000	2 010	6 110	4 100
C_1	1 001	3 011	7 111	5 101

	$\bar{C}_0$	C_1
$a\bar{b}$	0 000	1 001
$\bar{a}b$	2 010	3 011
ab	6 110	7 111
$a\bar{b}$	4 100	5 101

4 Variables K-map:

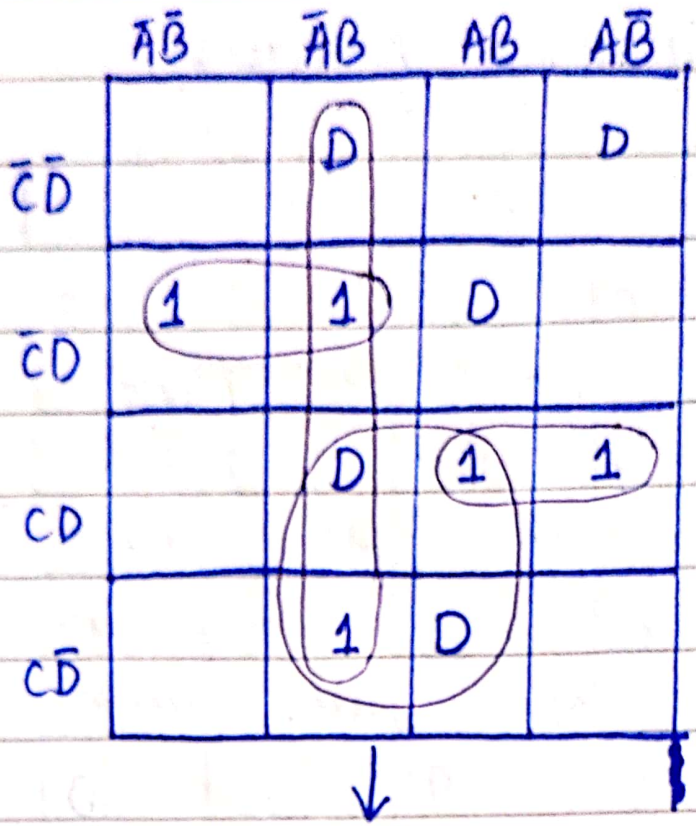
ab \ cd	00 $\bar{c}\bar{d}$	01 $\bar{c}d$	11 cd	10 $c\bar{d}$
$\bar{a}\bar{b}$ 00	0 0000	1 0001	3 0011	2 0010
$\bar{a}b$ 01	4 0100	5 0101	7 0111	6 0110
ab 11	12 1100	13 1101	15 1111	14 1110
$a\bar{b}$ 10	8 1000	9 1001	11 1011	10 1010

Terms (Grouping)

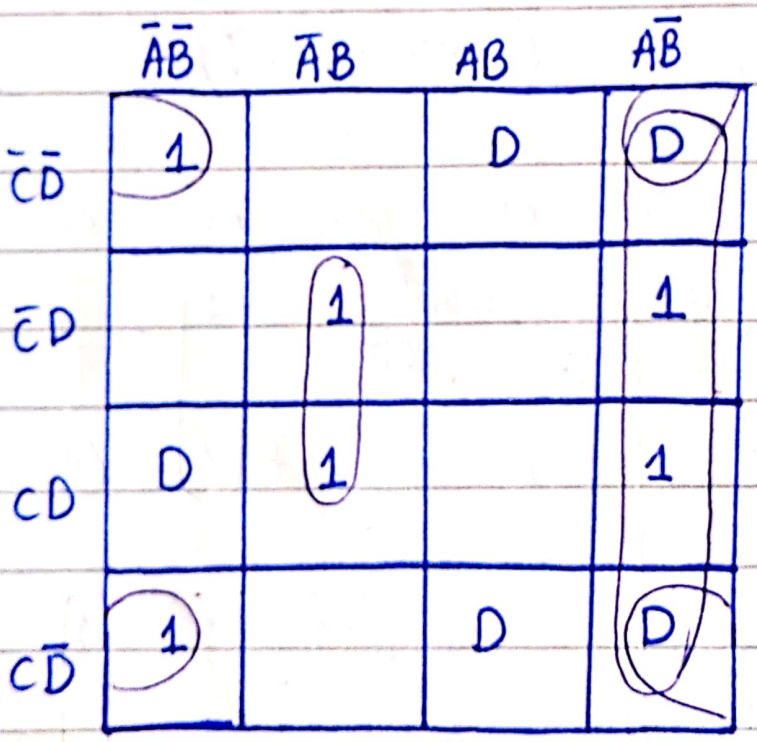


$BC+AB$

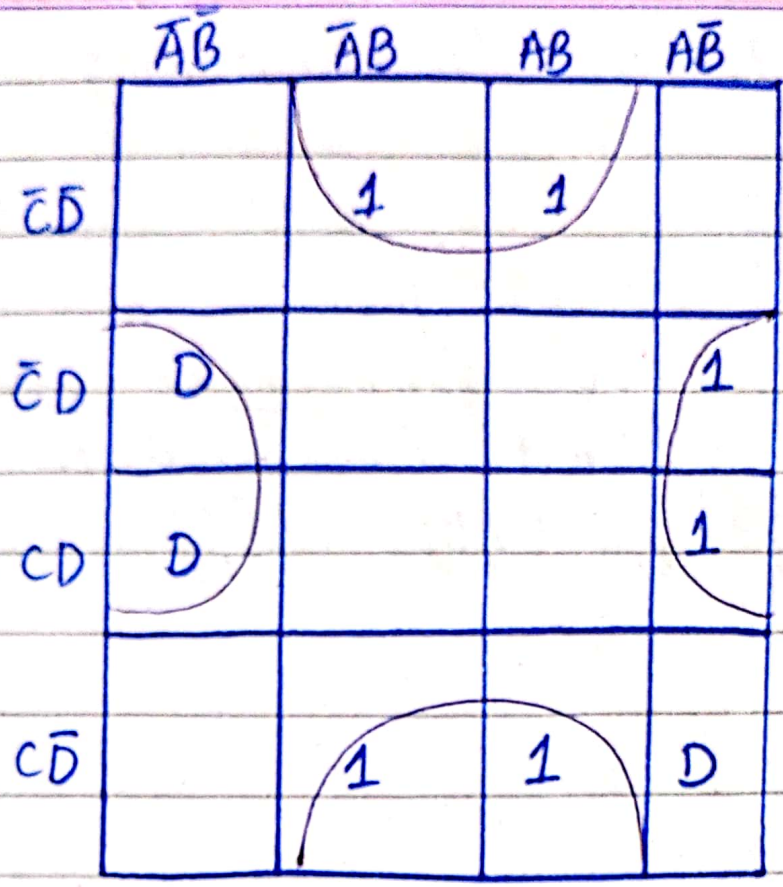
$\bar{A}CD+ABC+A\bar{C}D+\bar{A}B\bar{C}$



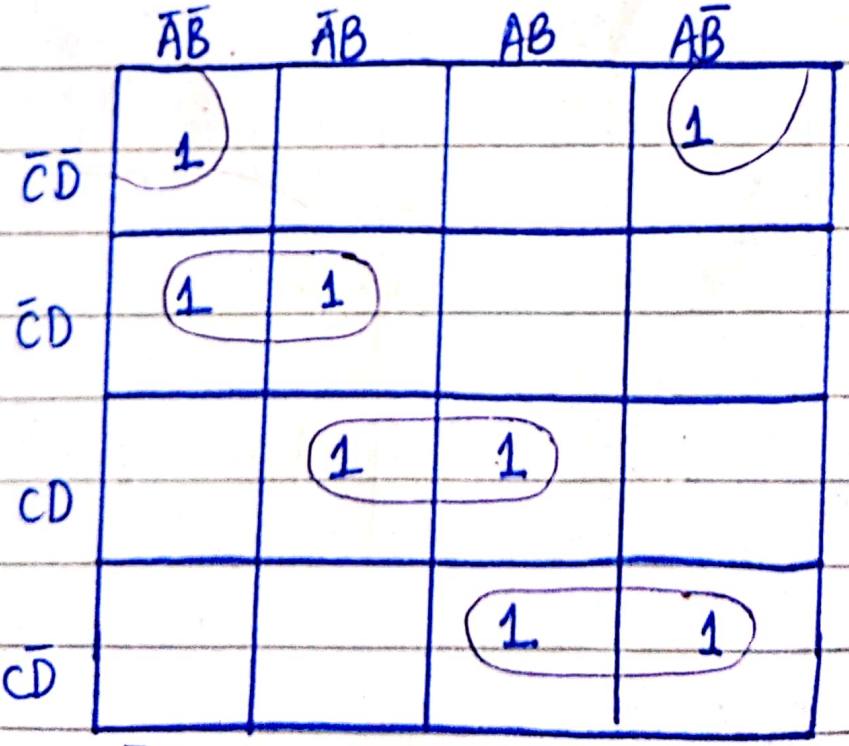
$$\bar{A}B + ACD + \bar{A}\bar{C}\bar{D} + BC + ACD + \bar{A}\bar{C}D$$



$$\bar{B}\bar{D} + \bar{A}BD + A\bar{B}$$



$$B\bar{D} + \bar{B}D = B \oplus D$$



$$\bar{A}\bar{C}D + BCD + AC\bar{D} + \bar{B}\bar{C}\bar{D}$$

جس بیر بار (-) ہوگا اسکی Value (0) ہوگی اور بار کے بغیر (1) ہوگی۔

Using K-map to minimizing SOP expression.

Expand terms to Convert it into SSOP if necassary.

$\bar{A}\bar{B}\bar{C}\bar{D}$ (1000)

$\bar{A}B\bar{C}\bar{D}$ (0000)

$\bar{B}\bar{C}\bar{D}$ (0 0 0) + $\bar{A}B\bar{C}\bar{D}$ (0 1 0 0) + $A\bar{B}\bar{C}\bar{D}$ (1 0 0 0) + $\bar{A}\bar{B}CD$ (0 0 1 1) + $A\bar{B}CD$ (1 0 1 1) +

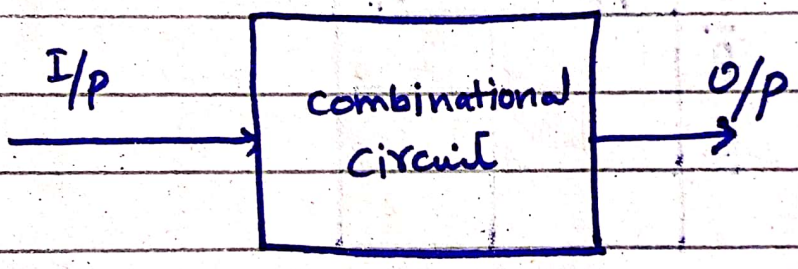
$\bar{A}BC\bar{D}$ (0 0 1 0) + $\bar{A}BC\bar{D}$ (0 1 1 0) + $ABC\bar{D}$ (1 1 1 0) + $A\bar{B}C\bar{D}$ (1 0 1 0)

15-March Final-term Lecture #01

• Combinational Circuit:

In digital electronics, combinational circuit is a circuit in which output depends on present combination of inputs. These circuits are made up of logic gates.

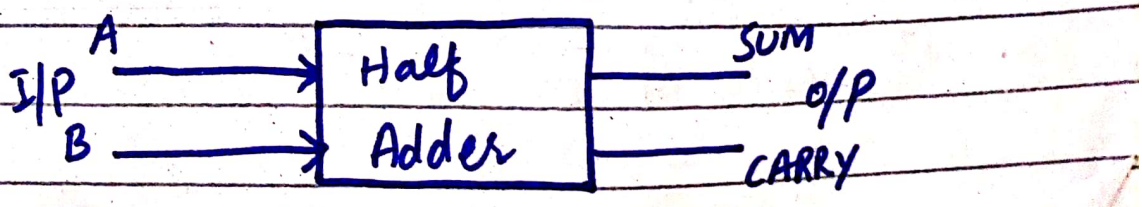
• Diagram:



1- Half adder:

It is a digital logic circuit that perform binary addition of two single bit binary numbers. It has two inputs **A** and **B**, two outputs **SUM** and **CARRY**.

• Diagram:



- Table:

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Table for Sum:

SUM

	$\bar{A}$	A
$\bar{B}$	0	1 2
B	1 1	3

$SUM = \bar{A}B + A\bar{B} = A \oplus B$

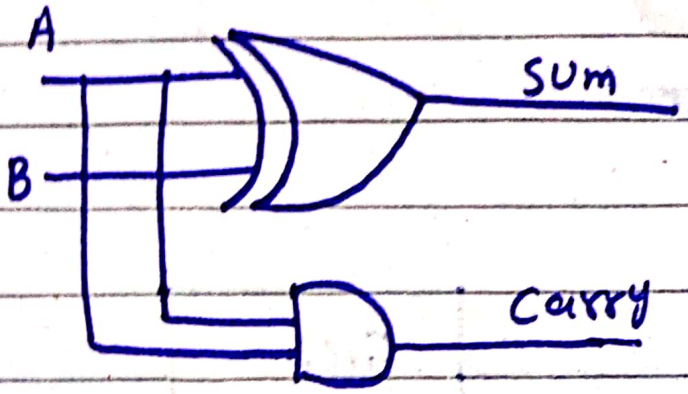
- Table for CARRY:

CARRY

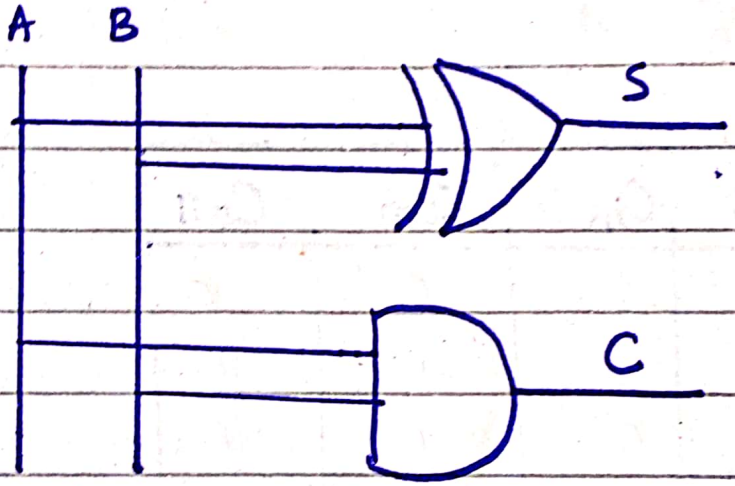
	$\bar{A}$	A
$\bar{B}$	0	2
B	1	3

$CARRY = AB$

Circuit Diagram:



Another method



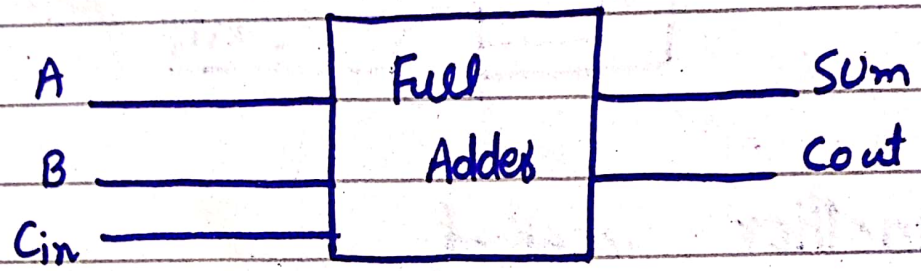
DLD Lecture#02

- **Full Adder:**

Full adder is the adder that adds three single bit number/ three inputs and produces two outputs. The first two ~~outputs~~ input are A and B

and the third input is carry (Cin)
The two outputs are Sum and Cout.

- **Diagram:**



- **Table:**

A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- **K-map:**

Sum

$\bar{C}in$	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
	0	1 (2)	6	1 (4)
Cin	1 (1)	3	1 (7)	5

$$Sum = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + AbC_{in} + A\bar{B}\bar{C}_{in}$$

$$= \bar{A}[\bar{B}C_{in} + B\bar{C}_{in}] + A[BC_{in} + \bar{B}\bar{C}_{in}]$$

$$= \bar{A}[B \oplus C_{in}] + A[B \odot C_{in}]$$

$$= \bar{A}[B \oplus C_{in}] + A[\overline{B \oplus C_{in}}]$$

$$= A \oplus B \oplus C_{in}$$

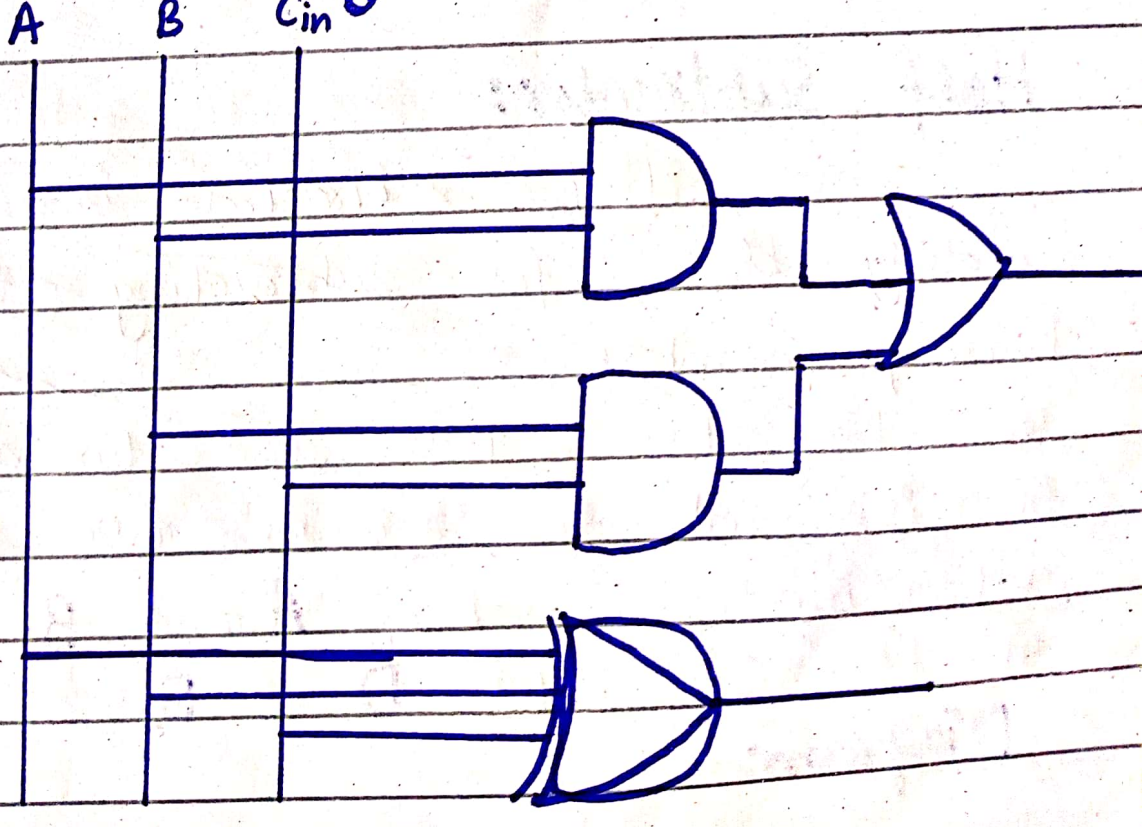
• K-map for Carry:

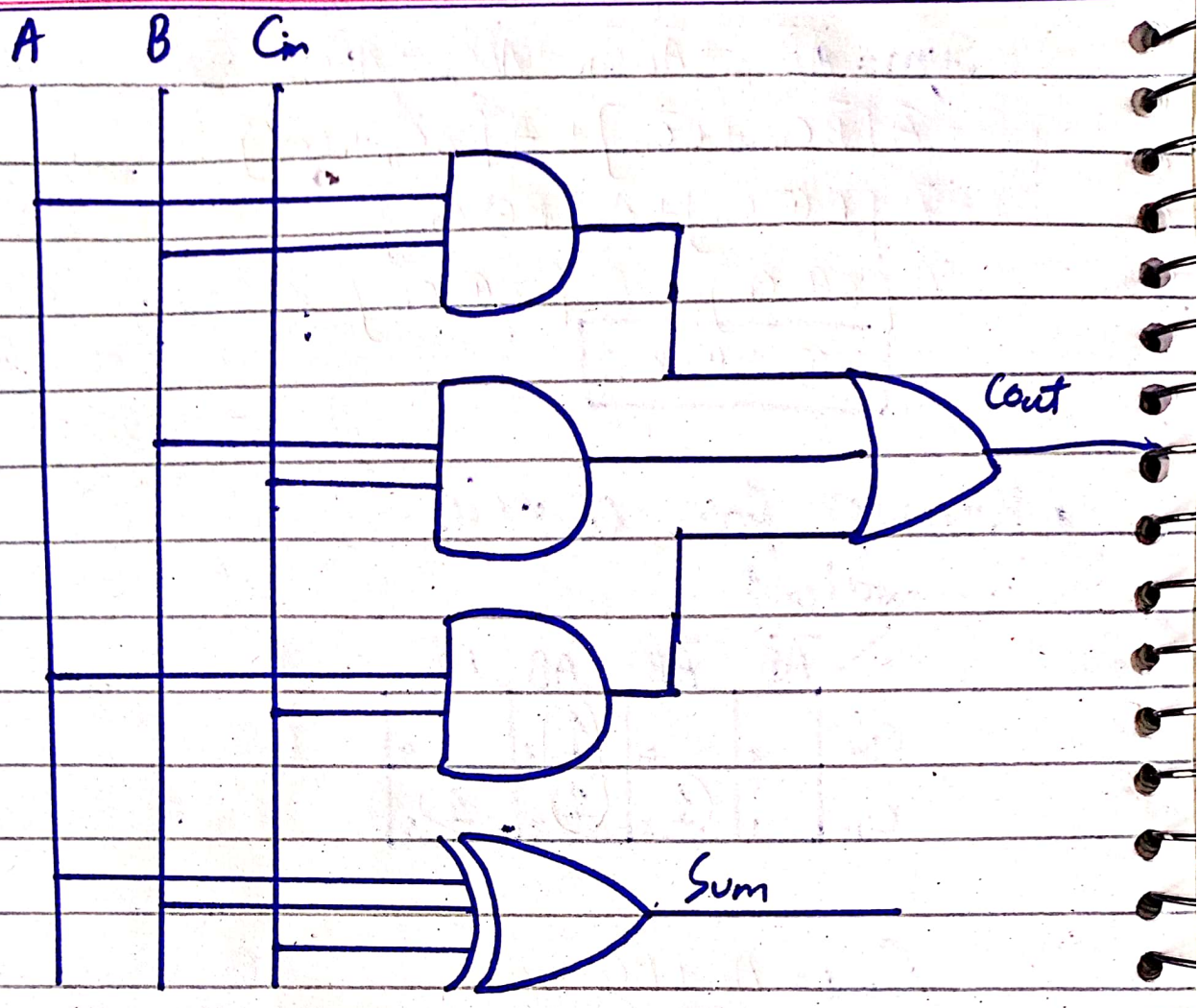
carry / Cout

Cin	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}_{in}$	0	2	1 (6)	4
C_{in}	1	1 (3)	1 (7)	1 (5)

$$C_{out} = AB + BC_{in} + AC_{in}$$

• Circuit Diagram:



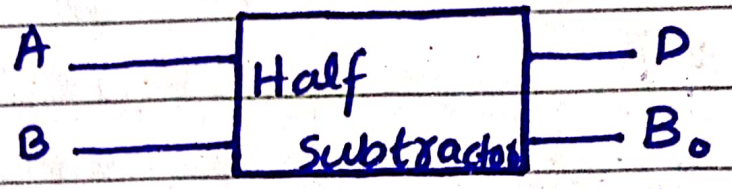


DLD Lecture#03

Half Subtractor:

Half subtractor is a building block for subtracting two binary numbers.
It has two inputs and two outputs.
It is used to subtract two single bit binary numbers A and B.
It has two output D and B.

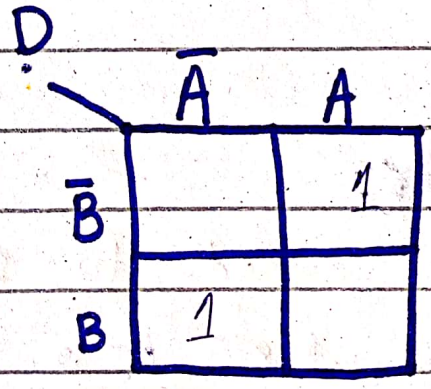
Diagram:



Truth table:

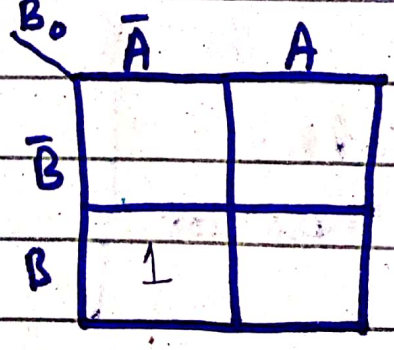
A	B	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map:



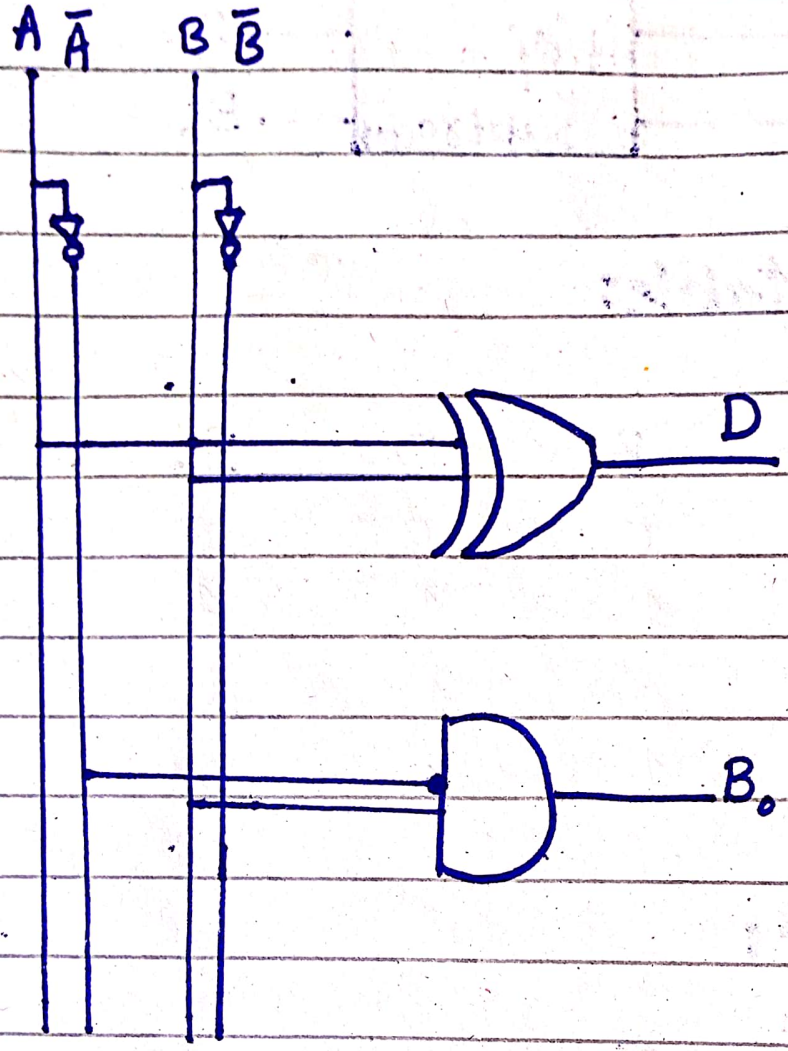
$D = A\bar{B} + \bar{A}B$

$D = A \oplus B$



$B_o = \bar{A}B$

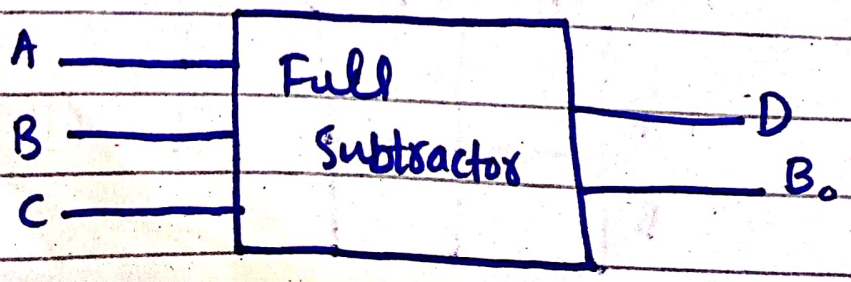
Circuit Diagram:



Full subtractor:

It is used to subtract three 1 bit numbers A, B & C. It has 3 inputs and 2 outputs.

Diagram:



Truth Table:

A	B	C	D	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-map:

D

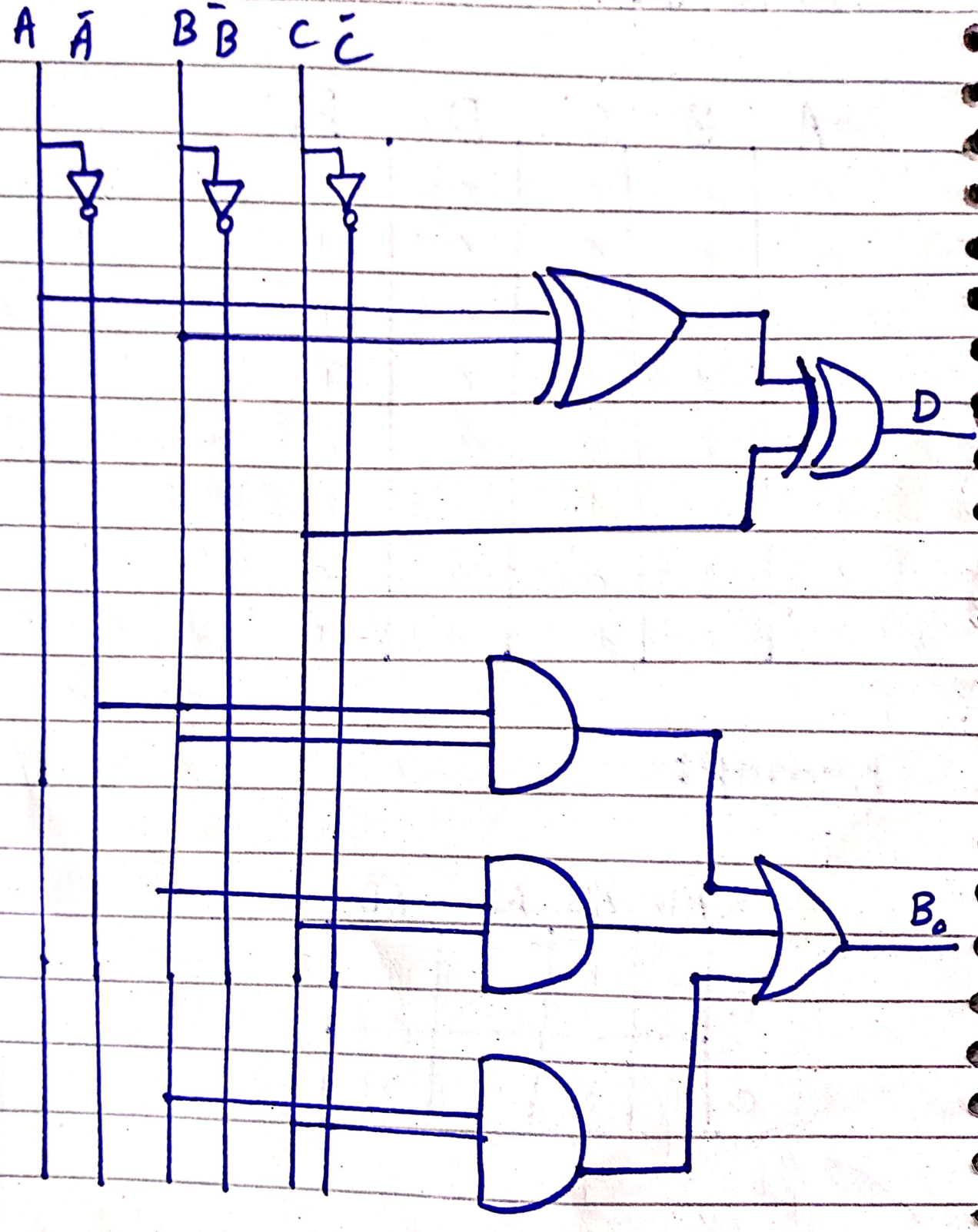
	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	0	1	0	1
C	1	0	1	0

$D = A \oplus B \oplus C$

B_0

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	0	1	0	0
C	1	1	1	0

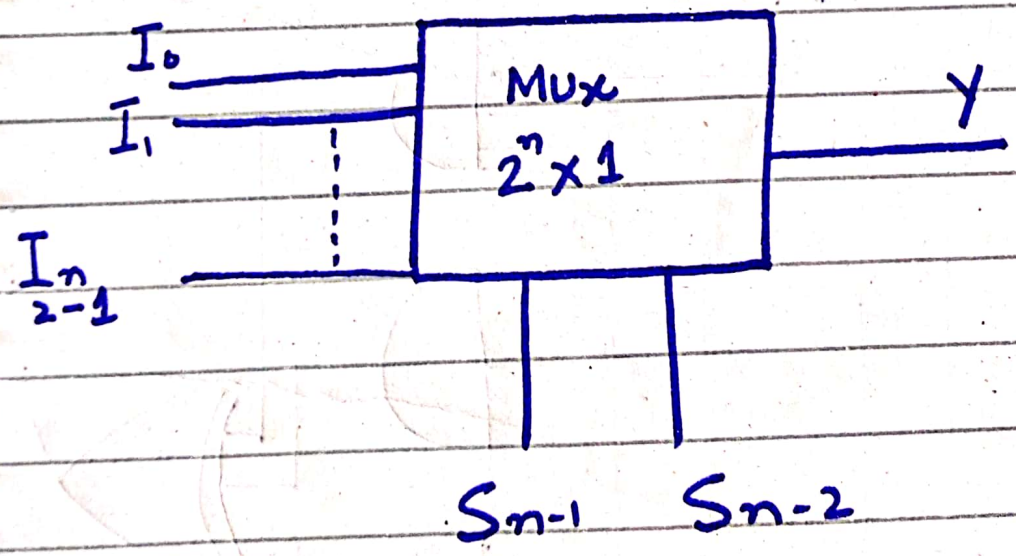
$B_0 = \bar{A}B + BC + \bar{A}C$


A
$\bar{A}$
B
$\bar{B}$
C
$\bar{C}$
D
B_0

Multiplexer (Mux)

- Also called data selector.
- widely used combination circuit.
- Here we select I/p from one of the many I/p lines (2^n) and direct it to a single O/p line.
- The selection of particular I/p line is controlled by a set of selection line.

Diagram:

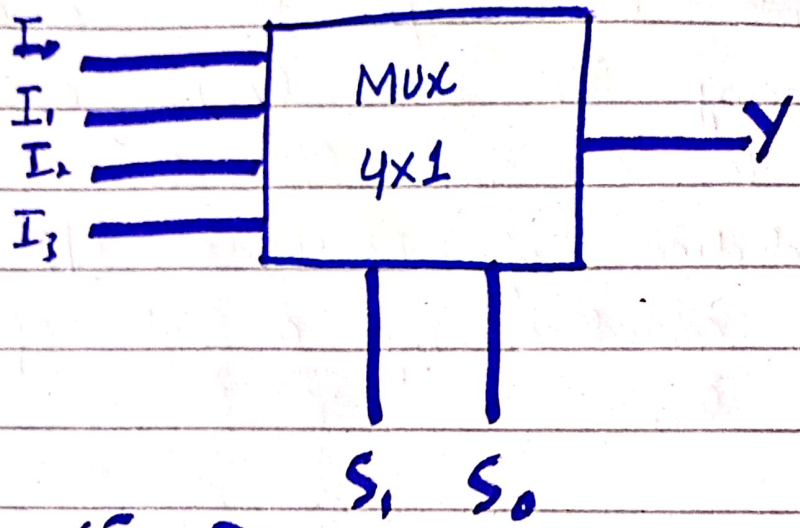


Truth table:

S_1	S_0	Y
0	0	I_0
0	1	I_1

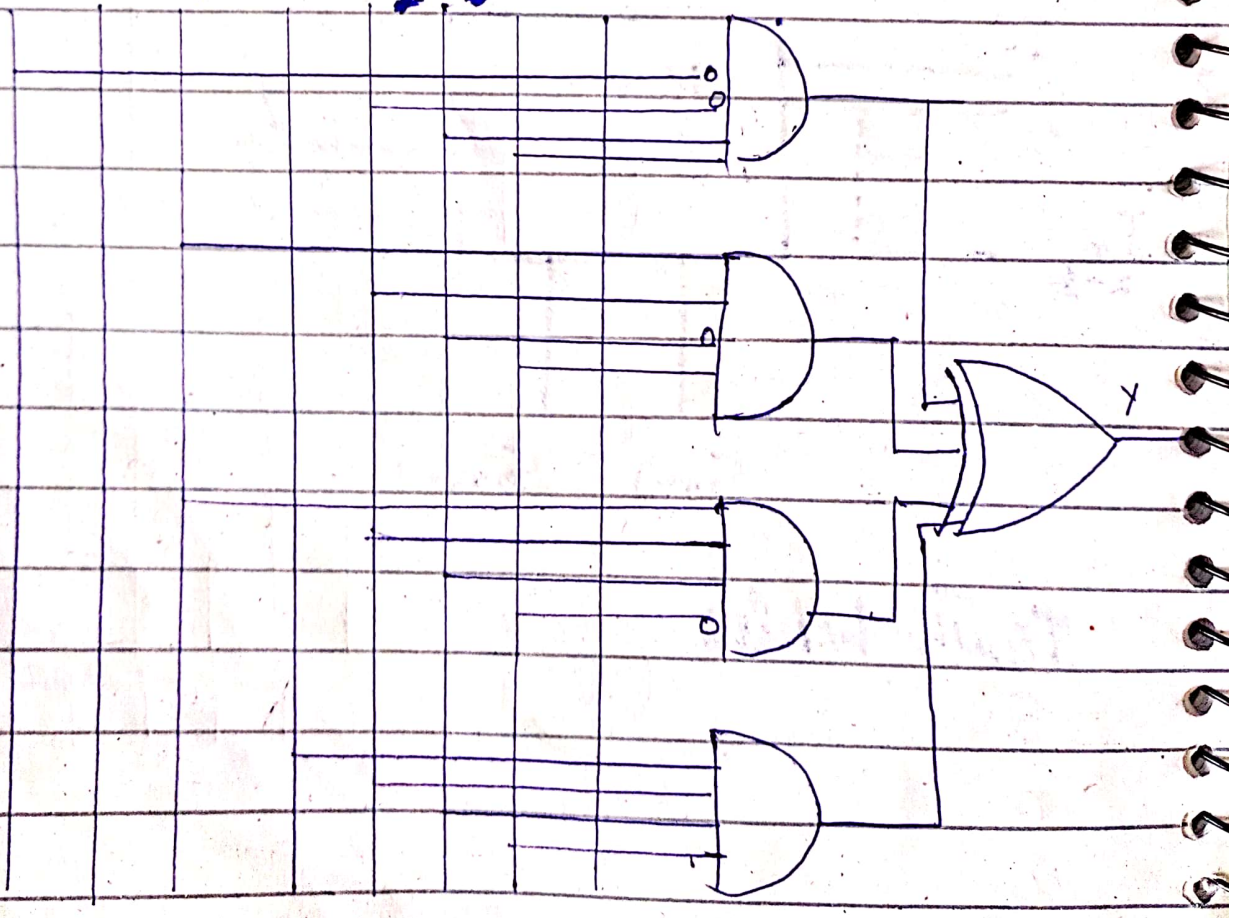
S_1	S_0	Y
1	0	I_2
1	1	I_3

Another Diagram:



Circuit Diagram:

I_0 I_1 I_2 I_3 S_1 S_0 E



Equation:

$$Y = E\left[\bar{S}_1 S_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3\right]$$

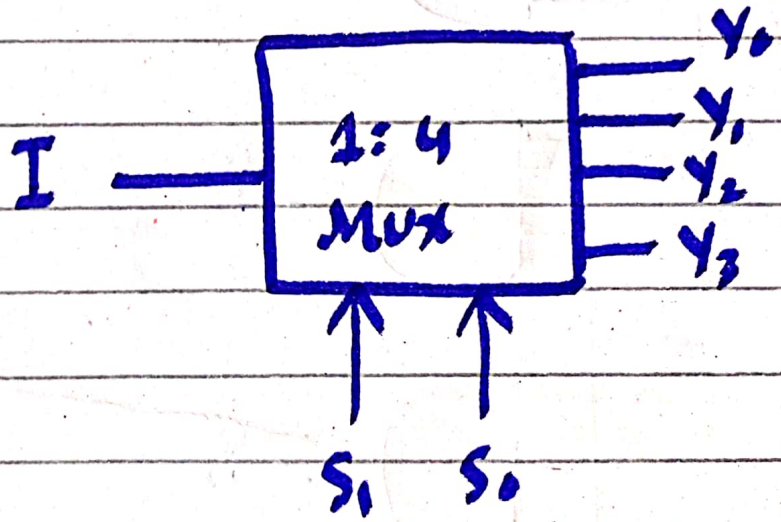
Demultiplexer (DeMux)

It is a combinational circuit that has one input line and 2^n output lines.

It is a single input and multi output combinational circuit.

It is also known as data distribution.

Diagram:



S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

Equations:

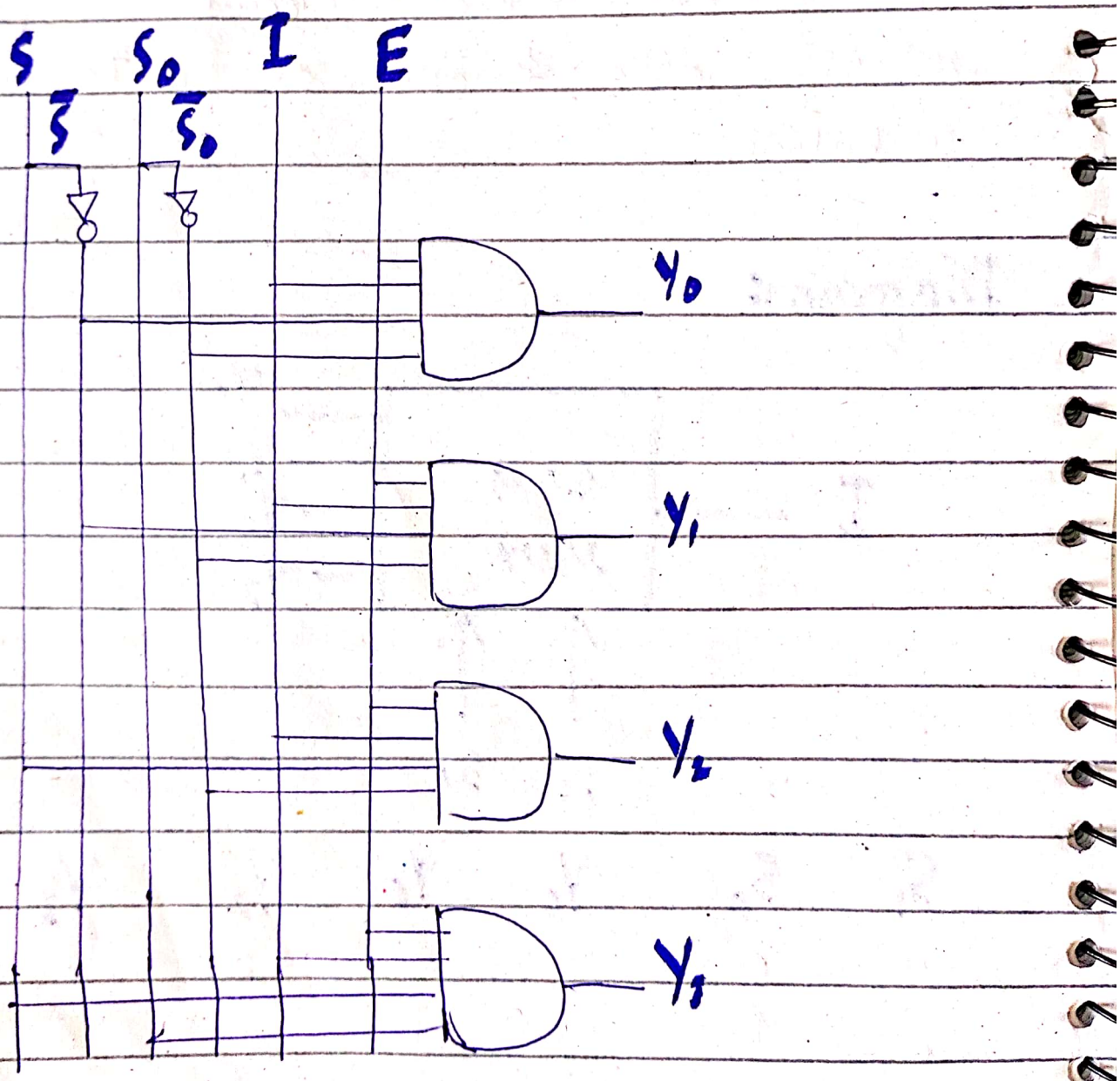
$Y_0 = E\bar{S}_1\bar{S}_0 I$

$Y_1 = E\bar{S}_1 S_0 I$

$Y_2 = ES_1\bar{S}_0 I$

$Y_3 = ES_1 S_0 I$

Circuit Diagram:



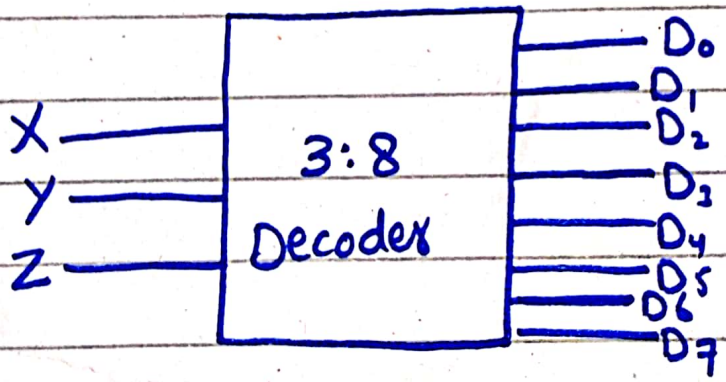
Decoder:

It is a multiple input and multiple output device. It is a Combinational circuit that Convert n input lines into 2^n (max) lines of output.

Applications of decoder are Converting:

1- Binary to Octal (3x8)
2- Binary to Hexa decimal (4x16)
3- Binary to Decimal (4x10)

3 to 8 lines Decoder:



Truth Table:

X	Y	Z	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0

X	Y	Z	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Equation:

$D_0 = \bar{X}\bar{Y}\bar{Z}$ $D_4 = X\bar{Y}\bar{Z}$

$D_1 = \bar{X}\bar{Y}Z$ $D_5 = X\bar{Y}Z$

$D_2 = \bar{X}Y\bar{Z}$ $D_6 = XY\bar{Z}$

$D_3 = \bar{X}YZ$ $D_7 = XYZ$

Encoder:

It is a combinational circuit that perform reverse operation of decoder. It has maximum of 2^n input lines and n output lines. Hence it encodes the information from 2^n input into n bit Code.

Truth Table:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	X	Y	Z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Expressions:

$$X = D_4 + D_5 + D_6 + D_7$$

$$Y = D_2 + D_3 + D_6 + D_7$$

$$Z = D_1 + D_3 + D_5 + D_7$$

→ Binary to grey code:

Dec	Binary	Grey code
+	0000	
	0001	
	0010	
	0011	
	0100	
	0101	
	0110	
	0111	
	1000	
	1001	
	1010	
	1011	
	1100	
	1101	
	1	

→ on next Page

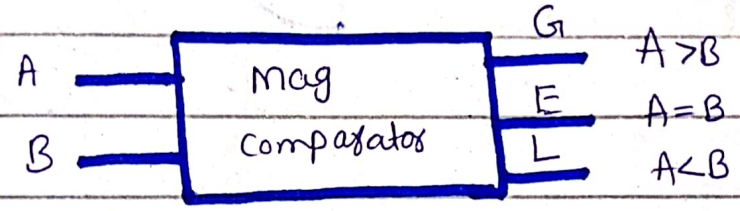
+

04th, 04, 2024

→ 1- Bit magnitude Comparator:

It is an electronic device that take two numbers as input in binary form and determined whether one number is greater than, less than or equal number to other numbers.

→ Diagram:



→ Truth Table:

A	B	G	E	L
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

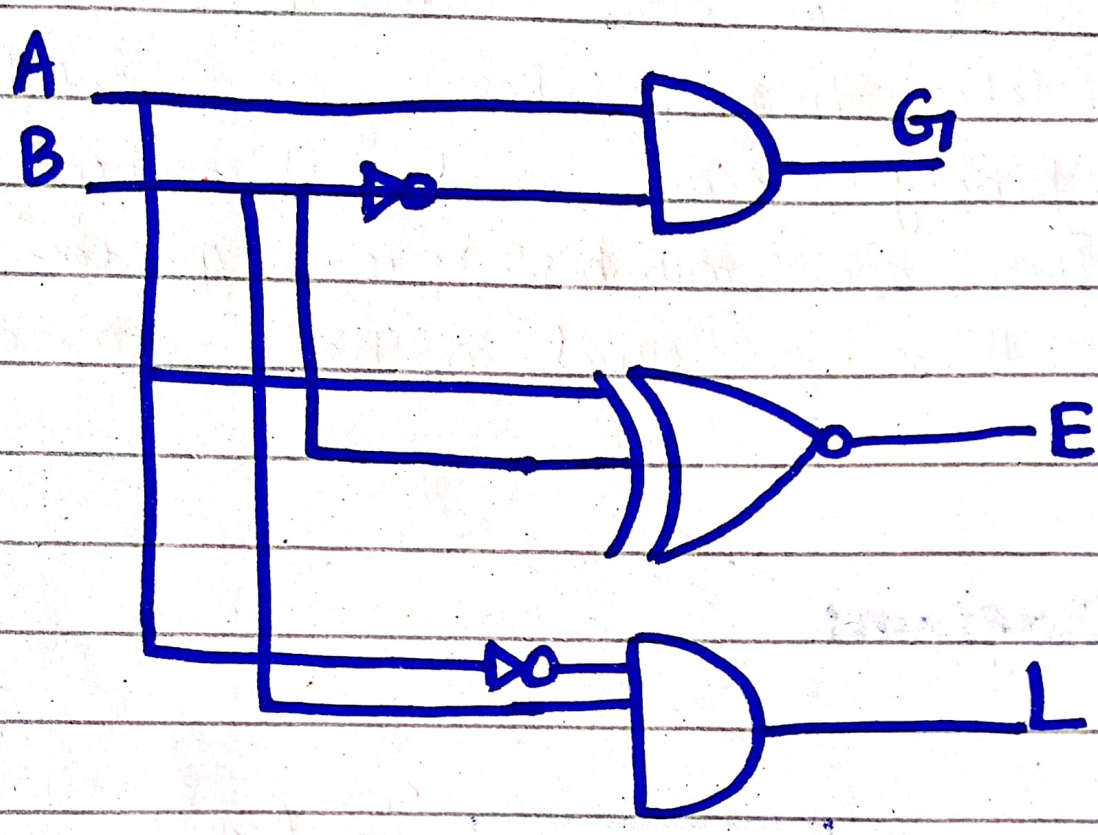
→ Expression:

$G = \bar{A}\bar{B}$

$E = \bar{A}\bar{B} + AB = A \odot B$

$L = \bar{A}B$

Circuit Diagram:



→ 2-Bit comparator:

A_2	A_1	B_2	B_1	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

→ K-map :

B_2B_1 \ A_2A_1	$\bar{A}_2\bar{A}_1$	$\bar{A}_2A_1$	A_2A_1	$A_2\bar{A}_1$
$\bar{B}_2\bar{B}_1$	0	1 (4)	1 (12)	1 (8)
$\bar{B}_2B_1$	1	5	1 (13)	1 (9)
B_2B_1	3	7	15	11
$B_2\bar{B}_1$	2	6	1 (14)	10

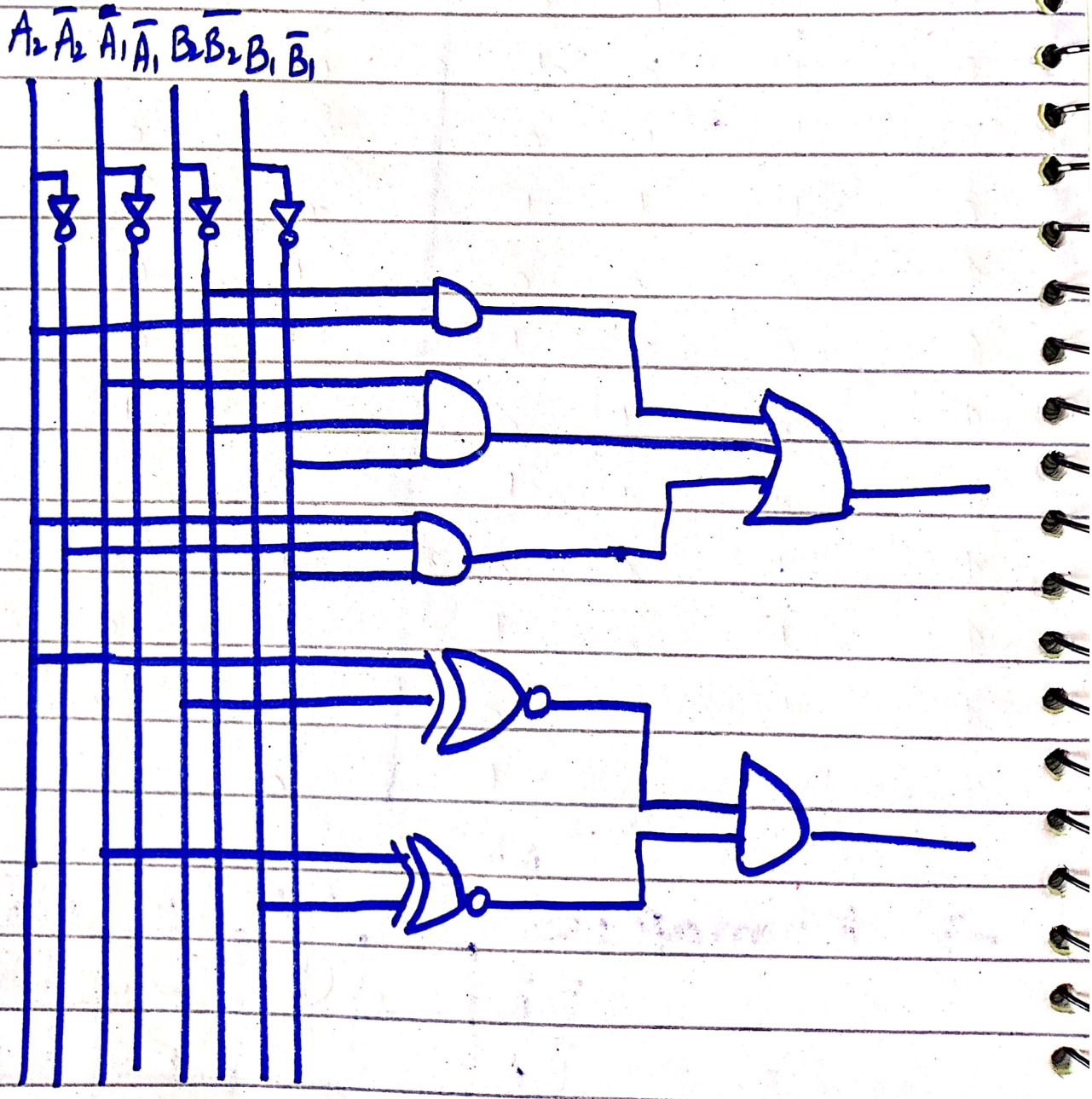
Expressions:-

$A > B = A_2 \bar{B}_2 + A_1 \bar{B}_2 \bar{B}_1 + A_2 A_1 \bar{B}$

$A = B : (A_2 \odot B_2)(A_1 \odot B_1)$

$A < B : \bar{A}_2 B_2 + \bar{A}_2 \bar{A}_1 B_1 + \bar{A}_1 B_2 B_1$

Circuit Diagram:

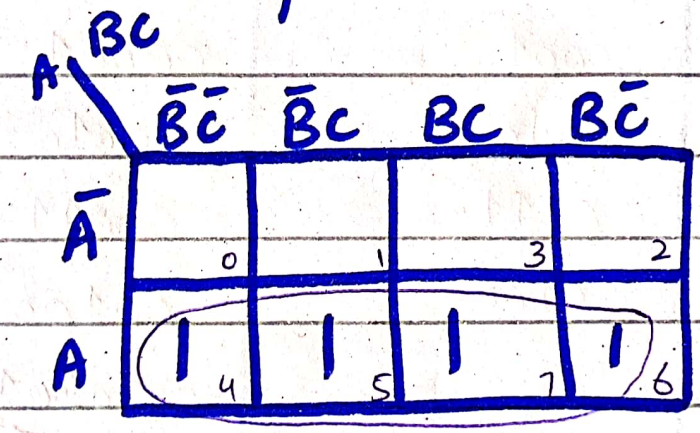


05-April-24

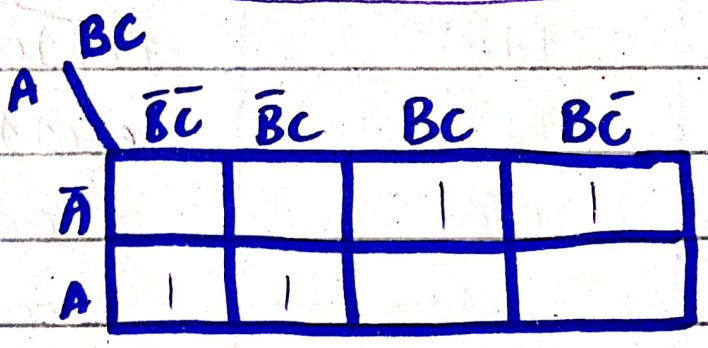
- **Binary to Grey code :**

Dec	Bin (A B C)	Grey (D E F)
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

- **K-map:**



D = A



$E = \bar{A}B + A\bar{B}$

$= A \oplus B$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$		1		1
A		1		1

$$F = \bar{B}C + B\bar{C}$$
$$= B \oplus C$$

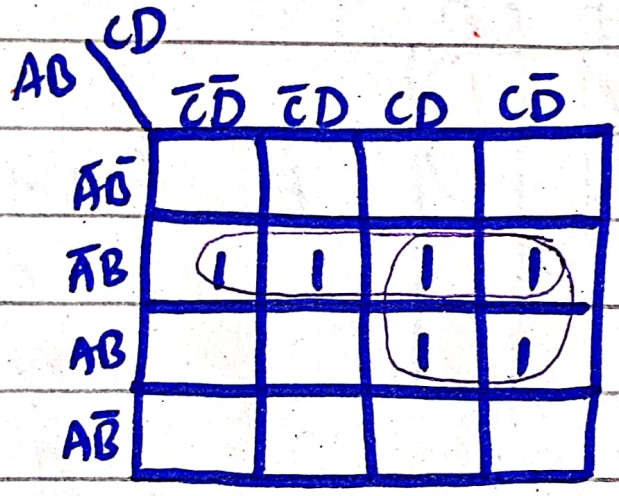
06-April-24

Binary to BCD Convertor:

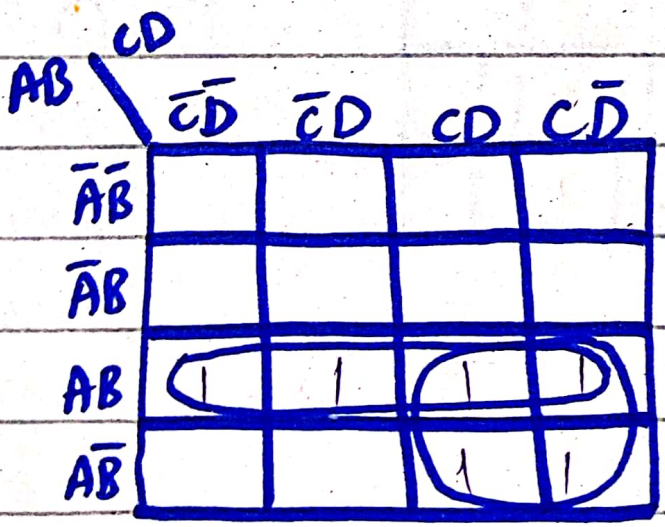
Dec	Bin ABCD	BCD EFGHI
0	0000	00000
1	0001	00001
2	0010	00010
3	0011	00011
4	0100	00100
5	0101	00101
6	0110	00110
7	0111	00111
8	1000	01000
9	1001	10000
10	1010	10001
11	1011	10010

12	1100	10010
13	1101	10011
14	1110	10100
15	1111	10101

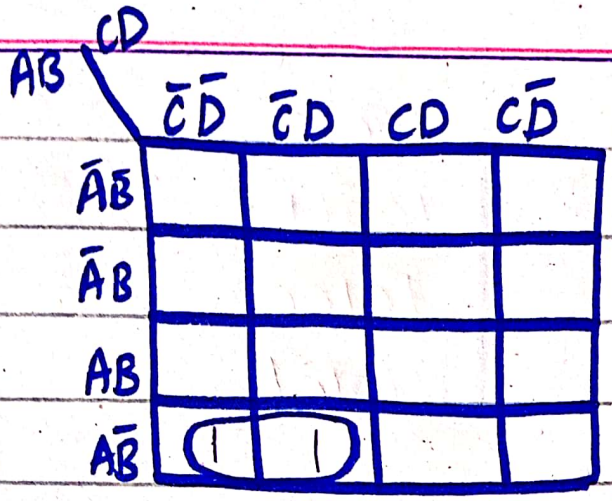
- **K-map:**



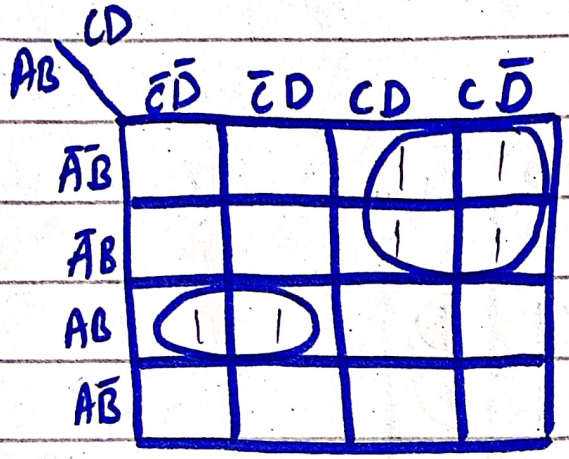
$G = BC + \bar{A}B$



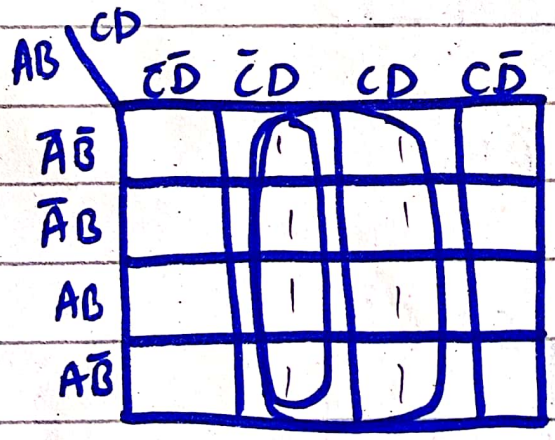
$E = AC + AB$



$F = A\bar{B}\bar{C}$



$H = \bar{A}C + AB\bar{C}$

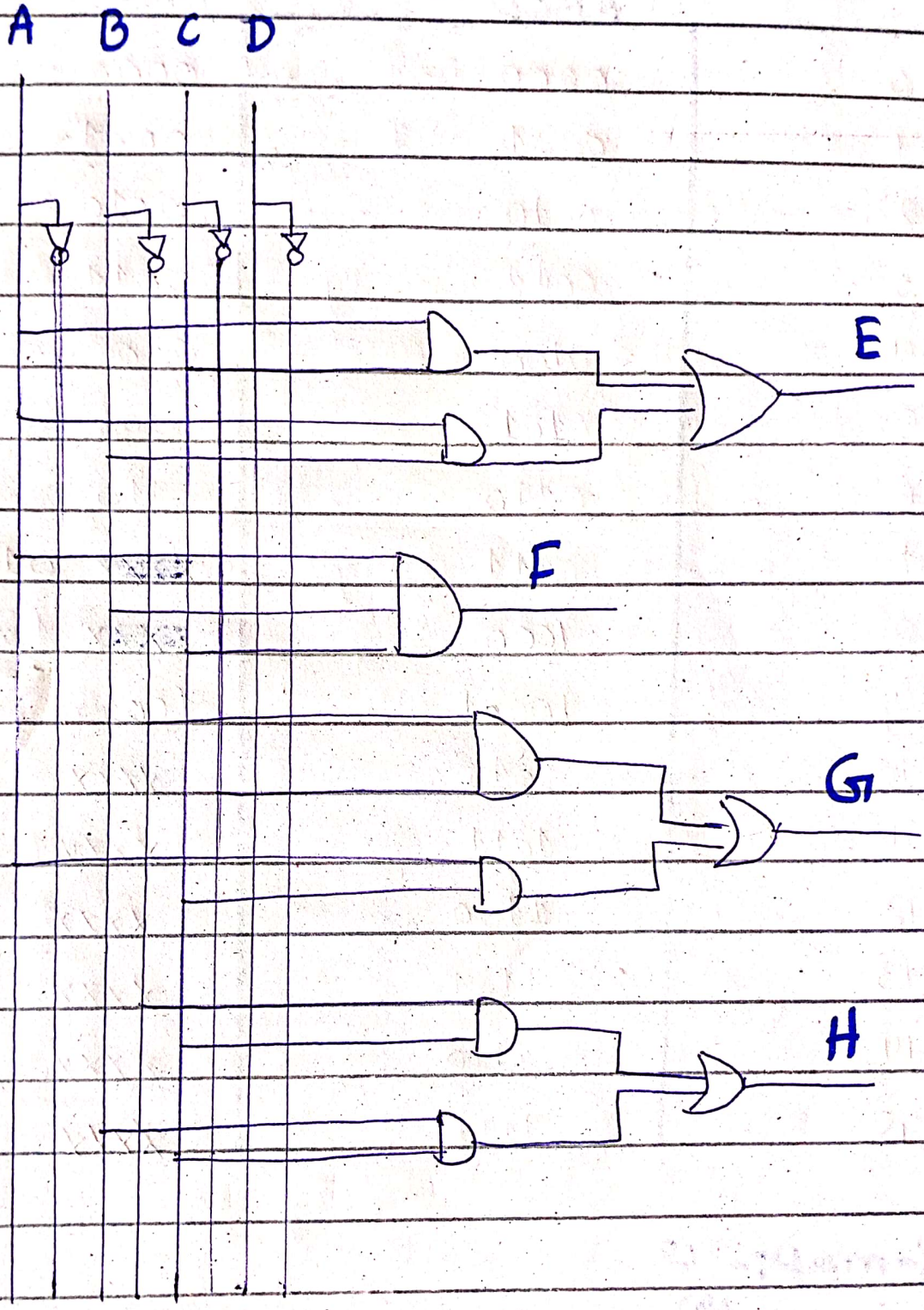


$I = D$

Circuit Diagram

→

• Circuit Diagram:

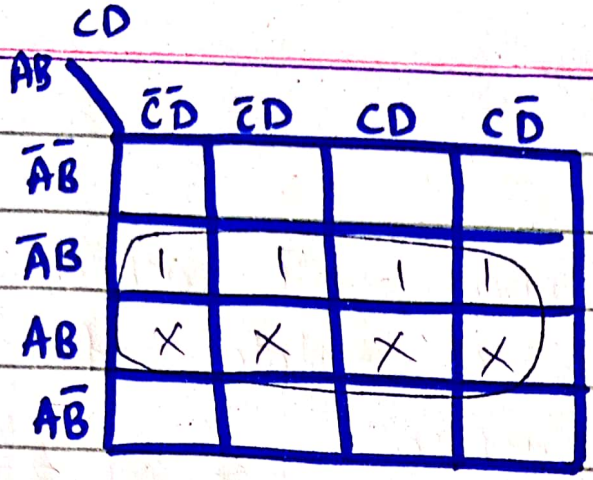


Dec	BCD ABCD	Bin PQRS
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	XXXX
11	1011	XXXX
12	1100	XXXX
13	1101	XXXX
14	1110	XXXX
15	1111	XXXX

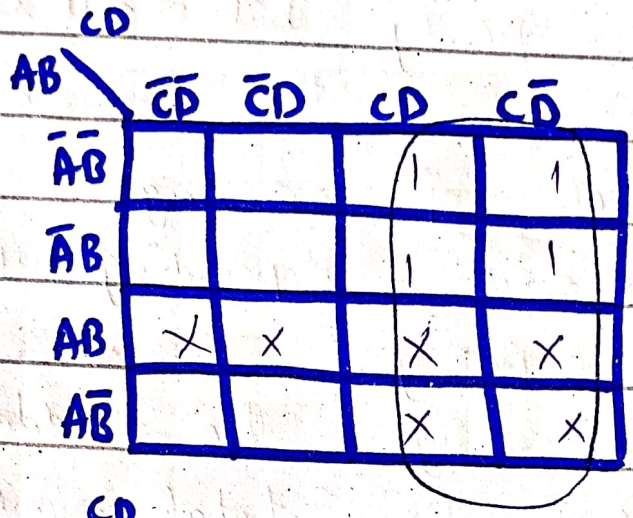
K-map:-

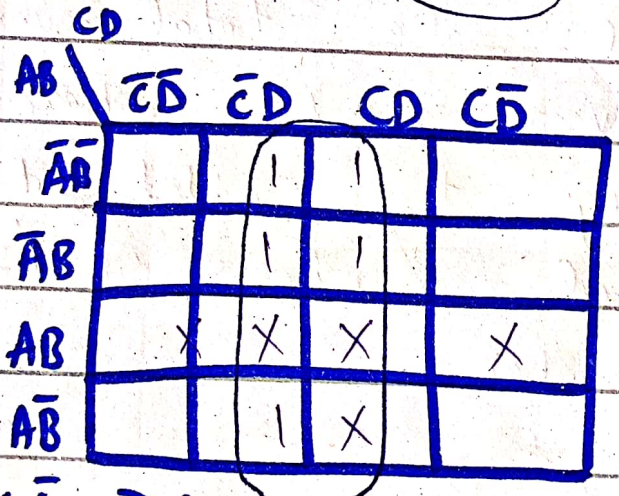
AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$				
AB	X	X	X	X
$A\bar{B}$	1	1	X	X

$P = A$

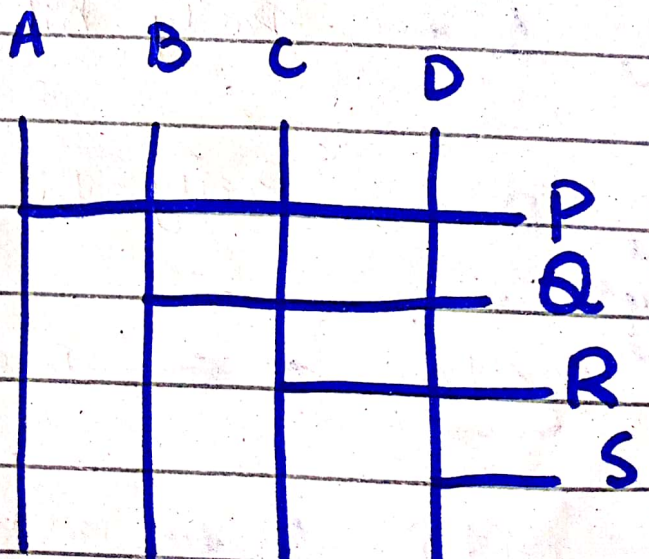


$Q = B$



$R = C$

$S = D$

Circuit Diagram:



19th April, 24

- **Parity Bit:**

→ is a check bit which is added to block of data for error detection purpose.

→ The value of parity bit either **0** or **1**, that makes the number of 1's in block of message either **even** or **odd** depending upon the Type of parity.

→ In even parity the added parity bit will make total number of is an even number whose as in odd parity, the added bit will make total number of 1's an add number.

- **Even parity:**

0010 **1** ← Parity bit

0110 **0** ← Parity bit

- **odd parity:**

0010 **0** ← Parity bit

0110 **1** ← Parity bit

• Parity generator and checker:

↳ are devices that help to ensure error free data transmission and processing in digital electronic system.

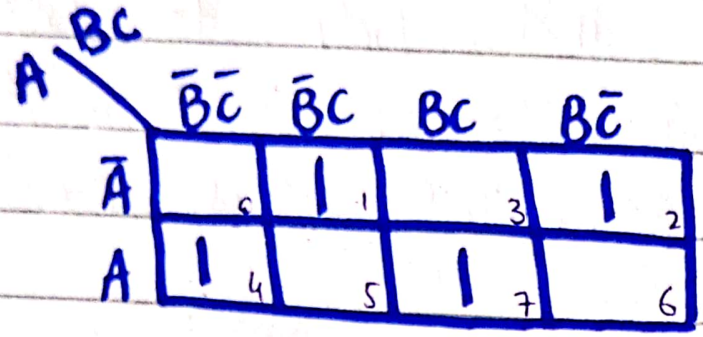
↳ A parity generator is a combinational logic circuit that generate parity bit in transmitter.

↳ On the other hand a circuit that checks the parity in the reciever is called **Parity Checker.**

Even Parity Generator:

A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

K-map:



P_{ev}: $A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$

$= \bar{B}(A\bar{C} + \bar{A}C) + B(AC + \bar{A}\bar{C})$

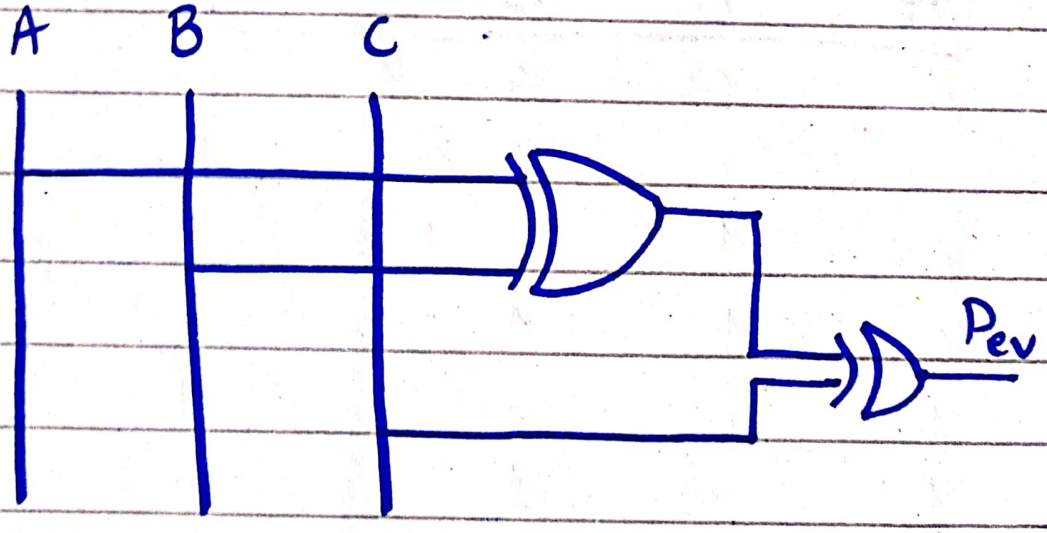
$= \bar{B}(A \oplus C) + B(A \odot C)$

$= \bar{B}(A \oplus C) + B\overline{(A \oplus C)}$

$= B \oplus A \oplus C$

$= A \oplus B \oplus C$

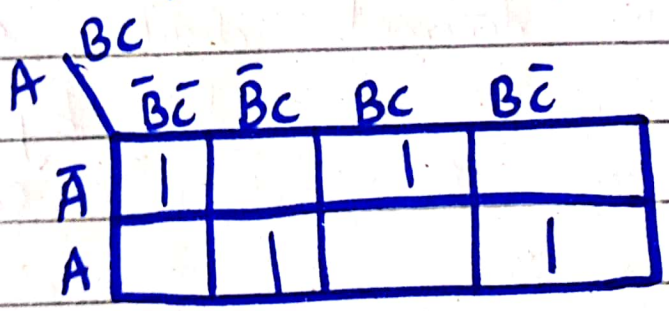
Circuit Diagram:



• Odd Parity generator:

A	B	C	P_{odd}
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

• K-map:



$$P_{odd} = \bar{A}\bar{B}\bar{C} + A\bar{B}C + \bar{A}BC + AB\bar{C}$$

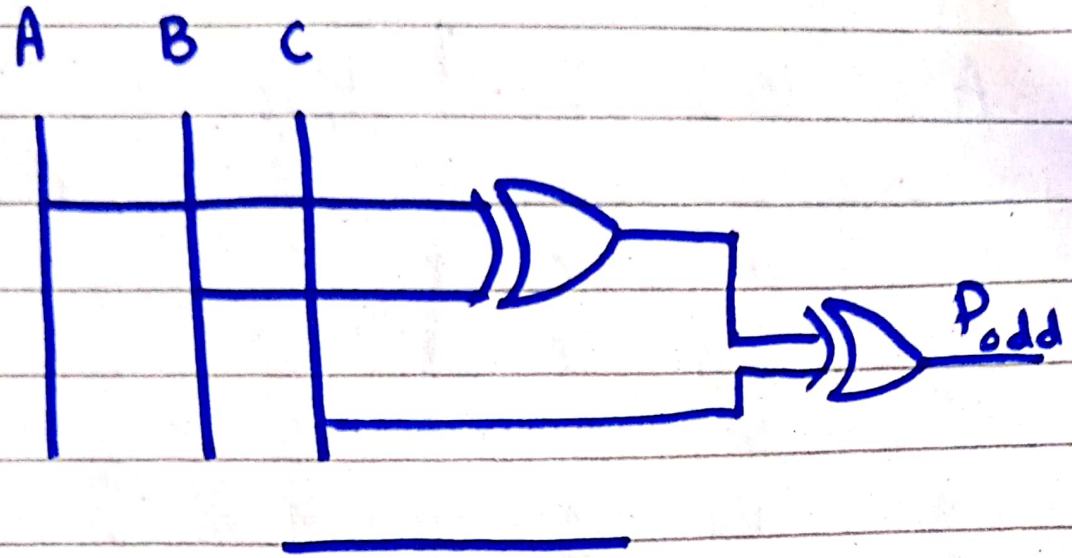
$$= \bar{C}(AB + \bar{A}\bar{B}) + C(A\bar{B} + \bar{A}B)$$

$$= \bar{C}(A \odot B) + C(A \oplus B)$$

$$= \bar{C}\,\overline{(A \oplus B)} + C(A \oplus B)$$

$$= C \odot (A \oplus B)$$

- Circuit Diagram:



DLD

- Excess-3 code:

In excess-3 code, numbers are represented as decimal digits, and each digit is represented by four bits as the digit value by plus 3.

- Table:

Decimal	BCD	Excess-3
0	0000	0011
1	0001	0100
2	0010	0101

3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

• Advantages of excess-3 code:

1- The excess-3 code is non-weighted code used to express code used as decimal numbers.

2- It is a self-complimentary binary coded decimal.

3- They use biased representation.

• Example:

2 → 0010 + 0011 = 0101

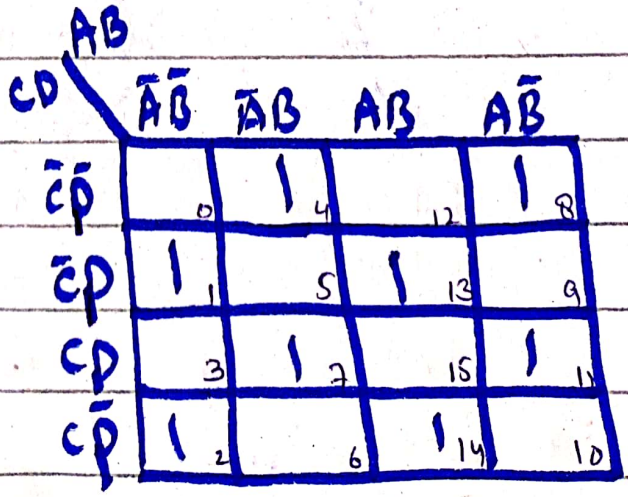
6 → 0110 + 0011 = 1001

0101 1001

- Even-parity checker:

A	B	C	D	P_e
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

- K-map:



• Expression:

$$P_{ev} = \bar{A}\bar{B}\bar{C}P + \bar{A}\bar{B}C\bar{P} + \bar{A}B\bar{C}\bar{P} + \bar{A}BCP + A\bar{B}\bar{C}\bar{P} + A\bar{B}CP + AB\bar{C}P + ABC\bar{P}$$

$$= \bar{A}\bar{B}(\bar{C}P + C\bar{P}) + \bar{A}B(\bar{C}\bar{P} + CP) + A\bar{B}(\bar{C}\bar{P} + CP) + AB(\bar{C}P + C\bar{P})$$

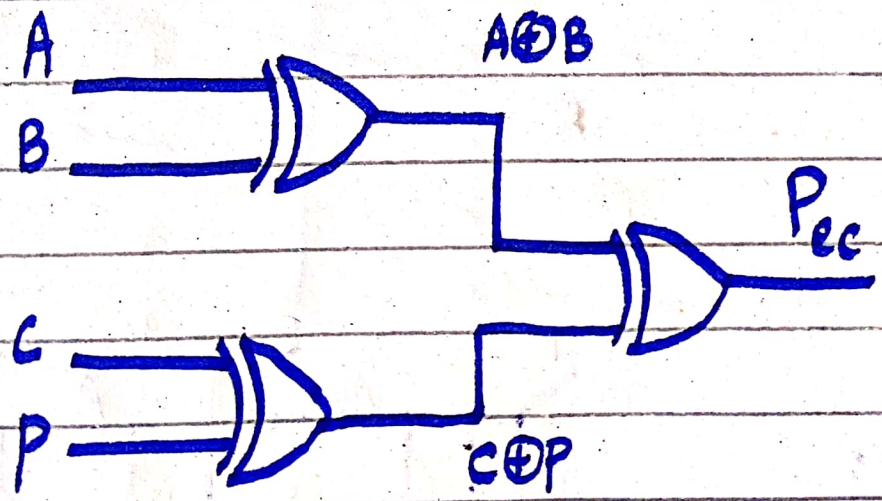
$$= \bar{A}\bar{B}(C \oplus P) + \bar{A}B(\overline{C \oplus P}) + A\bar{B}(\overline{C \oplus P}) + AB(C \oplus P)$$

$$= (C \oplus P)(\bar{A}\bar{B} + AB) + (\overline{C \oplus P})(\bar{A}B + A\bar{B})$$

$$= (C \oplus P)(\overline{A \oplus B}) + (C \overline{\oplus} P)(A \oplus B)$$

$$\boxed{P_{ec} = (A \oplus B) \overline{\oplus} (C \oplus P)}$$

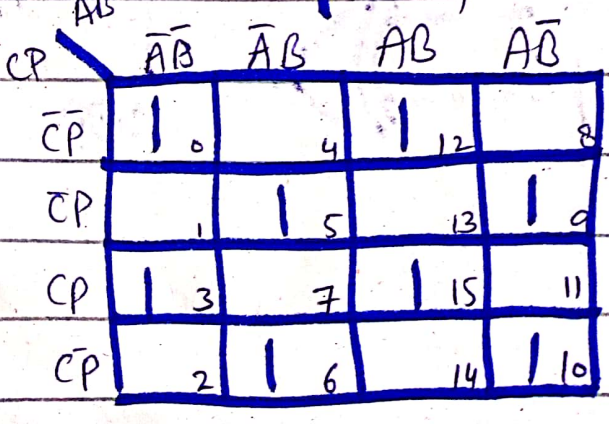
• Circuit Diagram:



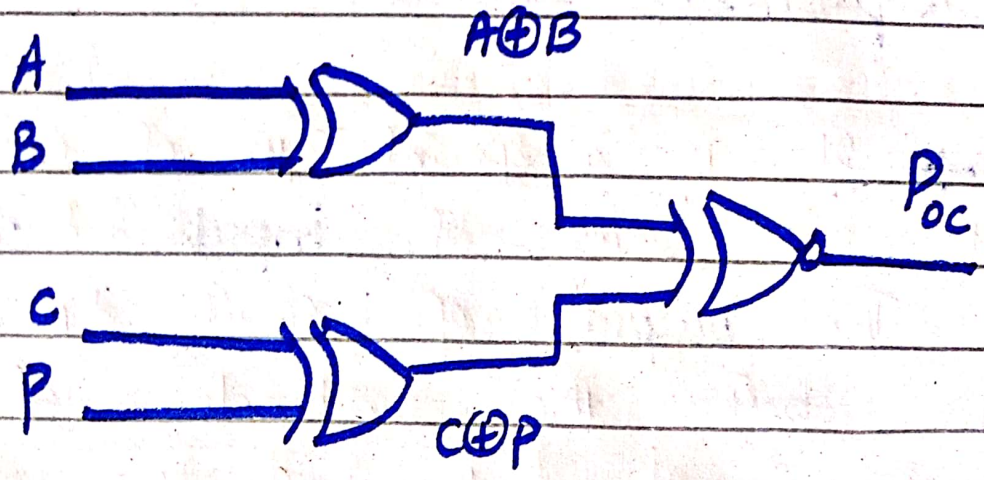
• Odd Parity Checker:

A	B	C	P	P_{OC}
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

• K-map:



- Circuit Diagram:



- expression:

$$P_{OC} = \bar{A}\bar{B}\bar{C}P + \bar{A}\bar{B}C\bar{P} + \bar{A}B\bar{C}P + \bar{A}BC\bar{P} + A\bar{B}\bar{C}P + A\bar{B}C\bar{P} + AB\bar{C}\bar{P} + ABCP$$

$$= \bar{A}\bar{B}(\bar{C}\bar{P} + CP) + \bar{A}B(\bar{C}P + C\bar{P}) + A\bar{B}(\bar{C}P + C\bar{P}) + AB(\bar{C}\bar{P} + CP)$$

~~$\bar{A}\bar{B}(\bar{C}\bar{P} + CP) + AB($~~

$$= \bar{A}\bar{B}(\overline{C \oplus P}) + \bar{A}B(C \oplus P) + A\bar{B}(C \oplus P) + AB(\overline{C \oplus P})$$

$$= (\overline{C \oplus P})(\bar{A}\bar{B} + AB) + (C \oplus P)(\bar{A}B + A\bar{B})$$

$$= (\overline{C \oplus P})(\overline{A \oplus B}) + (C \oplus P)(A \oplus B)$$

$$\boxed{P_{OC} = (A \oplus B) \odot (C \oplus P)}$$

27th April, 2024

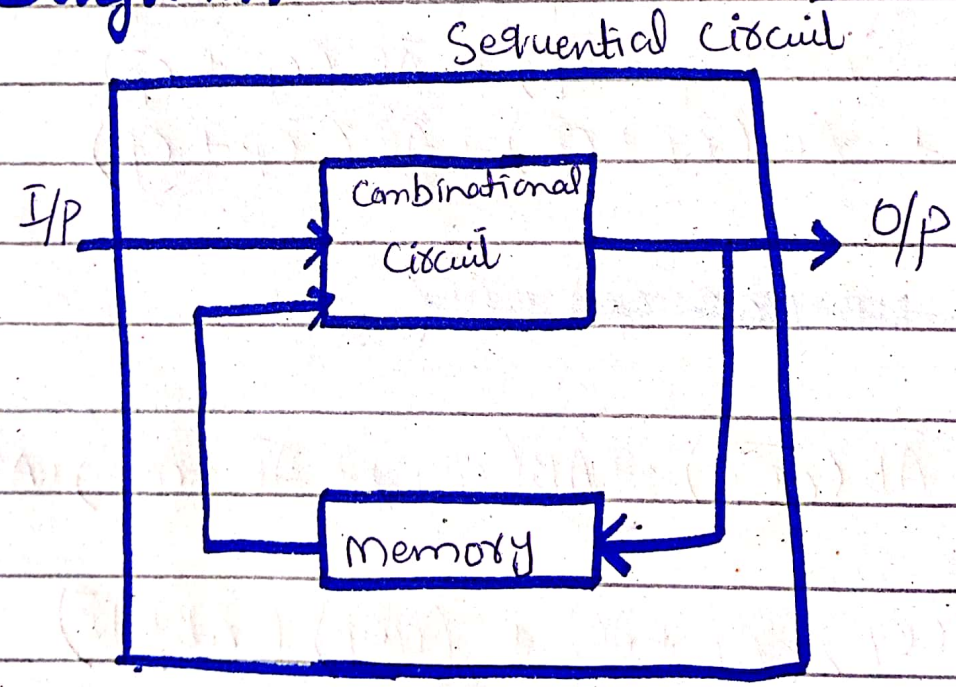
• Sequential Circuit:

→ It is a special type of circuit that has series of **inputs** and **outputs**

→ The **output** of circuit depends on both the combination of present input and previous output.

→ So, the Sequential circuit contains the combinational circuit and memory element.

• Diagram:



• LATCH:

* LATCH are basic building blocks using which **flip flop** are

0	0	1
0	1	1
1	0	1
1	1	0

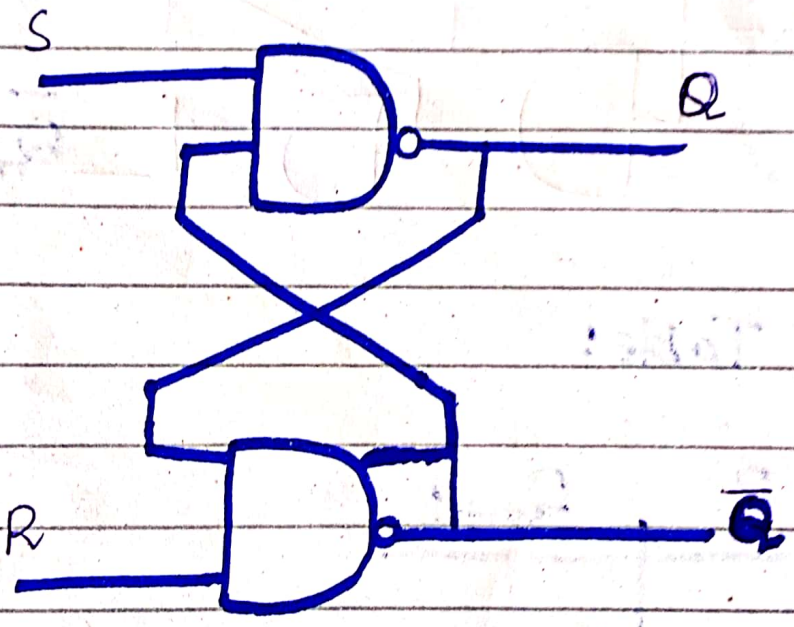
NAND gate

Constructed.

* They have capability to store **1 bit** until necessary.

LATCH

- **S-R Latch using NAND gate:**

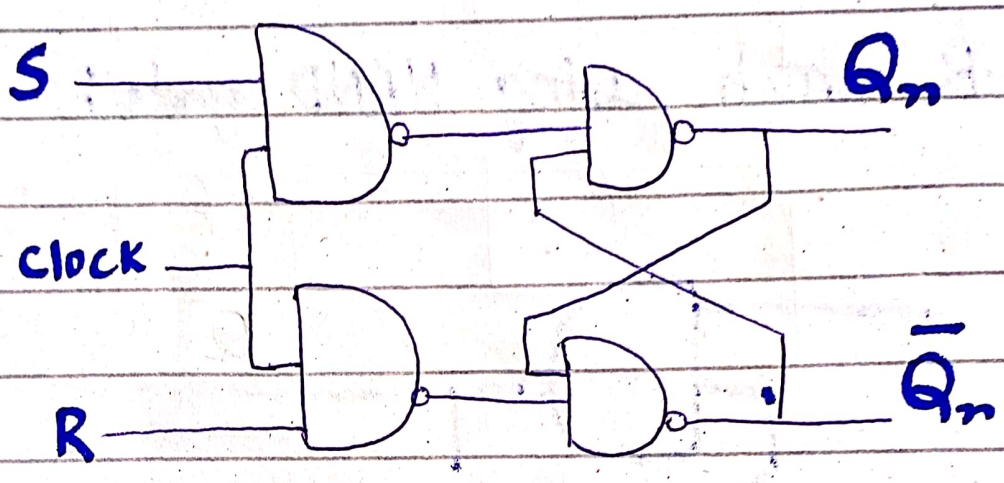


- **Table:**

S	R	Q_{n+1}
0	0	Invalid
0	1	1
1	0	0
1	1	Hold

- S-R Flip Flop:

- Circuit Diagram:



- Truth Table:

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	invalid

- **Characteristic Truth Table:**

S	R	Q_n	Q_{n+1}	
0	0	0	0	Latch mode
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	x	invalid
1	1	1	x	

- **K-map:**

	$\bar{S}\bar{R}$	$\bar{S}R$	SR	$S\bar{R}$
$\bar{Q}$	0	2	X 6	1 4
Q	1 1	3	X 7	1 5

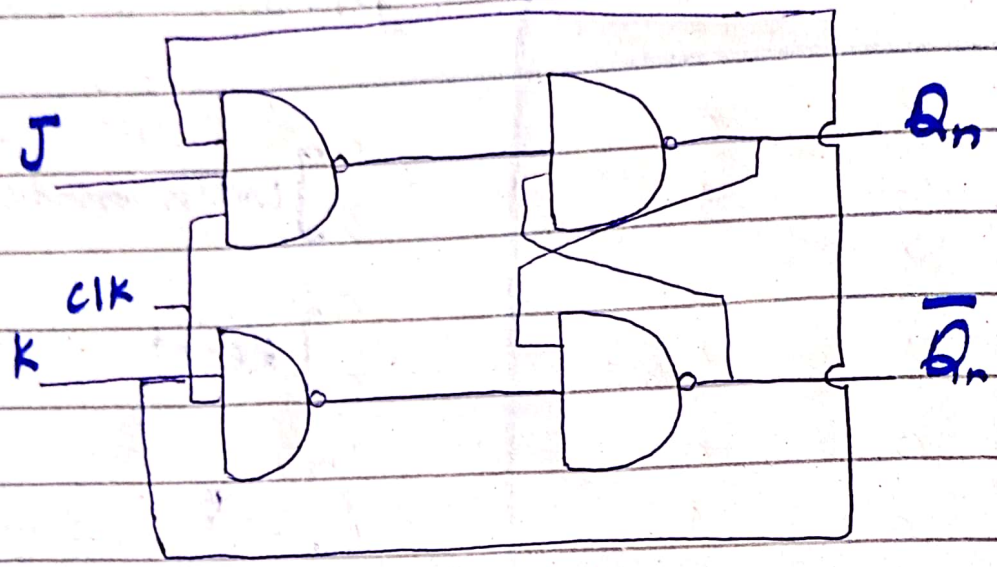
$$Q_{n+1} = S + \bar{R} Q_n$$

- **Excitation table:**

Q_n	Q_{n+1}	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

DLD

- J-K Flip Flop



-

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

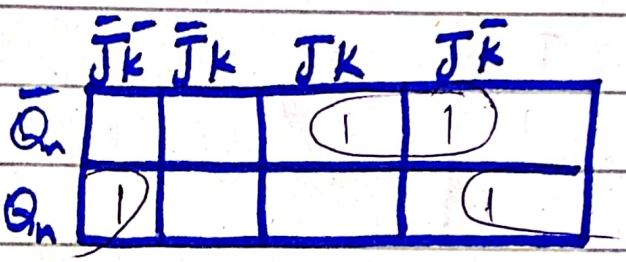
-

J	K	Q_n	Q_{n+1}	
0	0	0	0	Q_n
0	0	1	1	
0	1	0	0	Reset
0	1	1	0	
1	0	0	1	set
1	0	1	1	
1	1	0	1	$\bar{Q}_n$
1	1	1	0	

- **Excitation Table:**

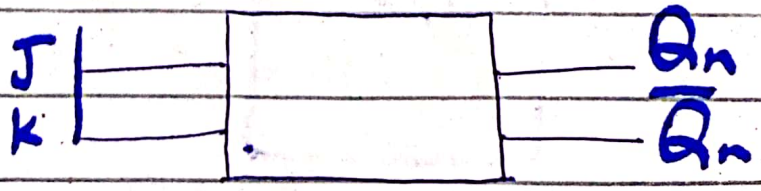
Q_n	Q_{n+1}	J	k
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

- **K-map:**



$$Q_{n+1} = J\bar{Q}_n + \bar{k}Q_n$$

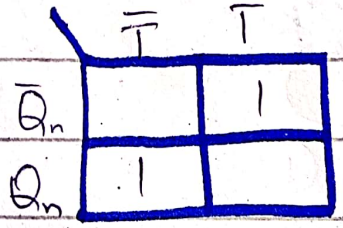
- **T- Flip Flop:**



- Truth Table:

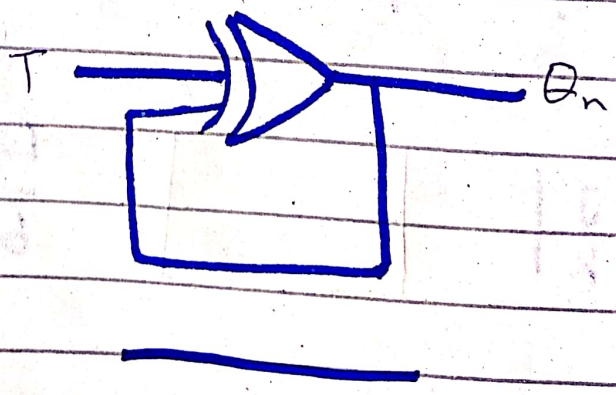
T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

- K-map

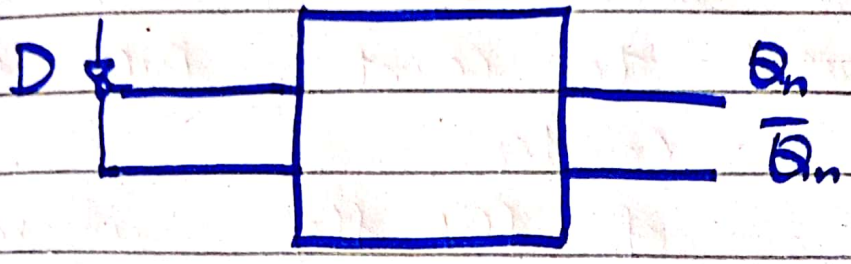


$$Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n$$
$$= T \oplus Q_n$$

- Circuit Diagram:



- D-Flip Flop



- Truth Table:

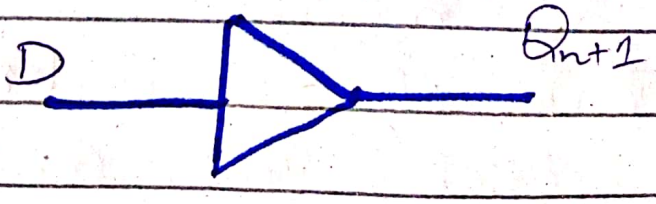
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

- K-map

	$\bar{D}$	D
$\bar{Q}_n$		1
Q_n		1

$Q_{n+1} = D$

- Circuit Diagram:



• Counters:

→ A special type of sequential circuit used to count pulse is known as Counter.

→ A collection of flip flops where clock signal is applied is known as **Counter.**

• Types:

There are following types of counters:

→ Asynchronous Counter

→ Synchronous Counter

• Asynchronous:

→ It is also known as ripple counter. In this counter the presented counter output passes to the input of the next counter.
So the counter are connected like a chain.

→ The drawback of this counter is that it creates the counting delay.

• Synchronous:

→ In this counter, same clock pulse is passed to the clock input of all flip flops.

→ The clock signals produced by all flips flop are the same.

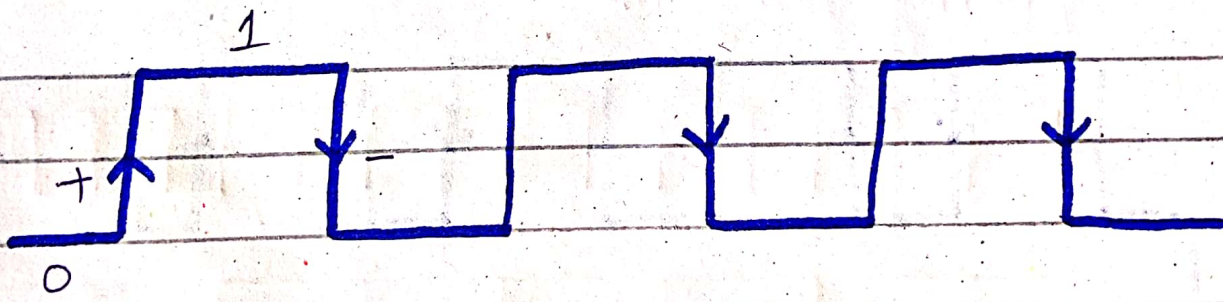
→ It is also known as synchronous.

• 3-bit Asynchronous up counter:

Counter start Trigger

1- Edge

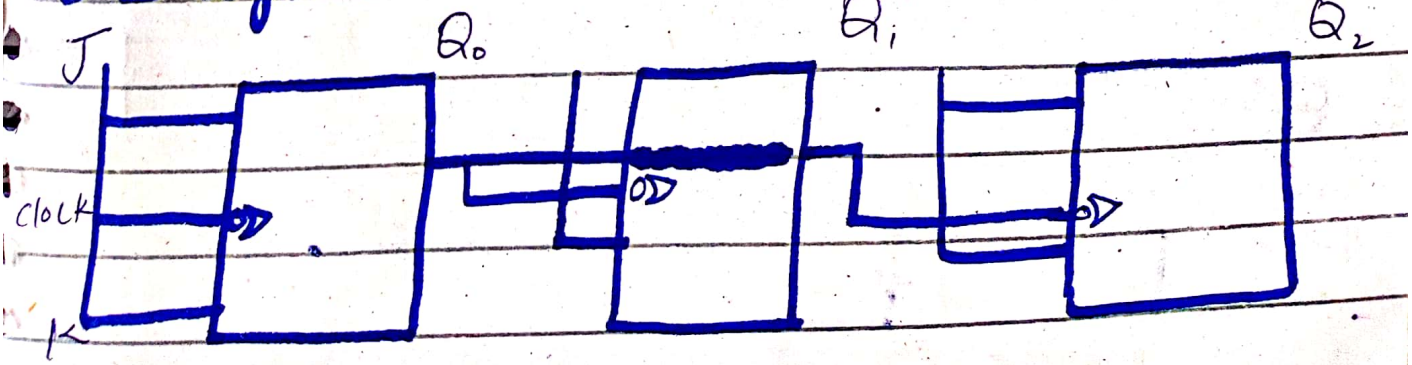
2- Level



‾‾ = level

| = edge

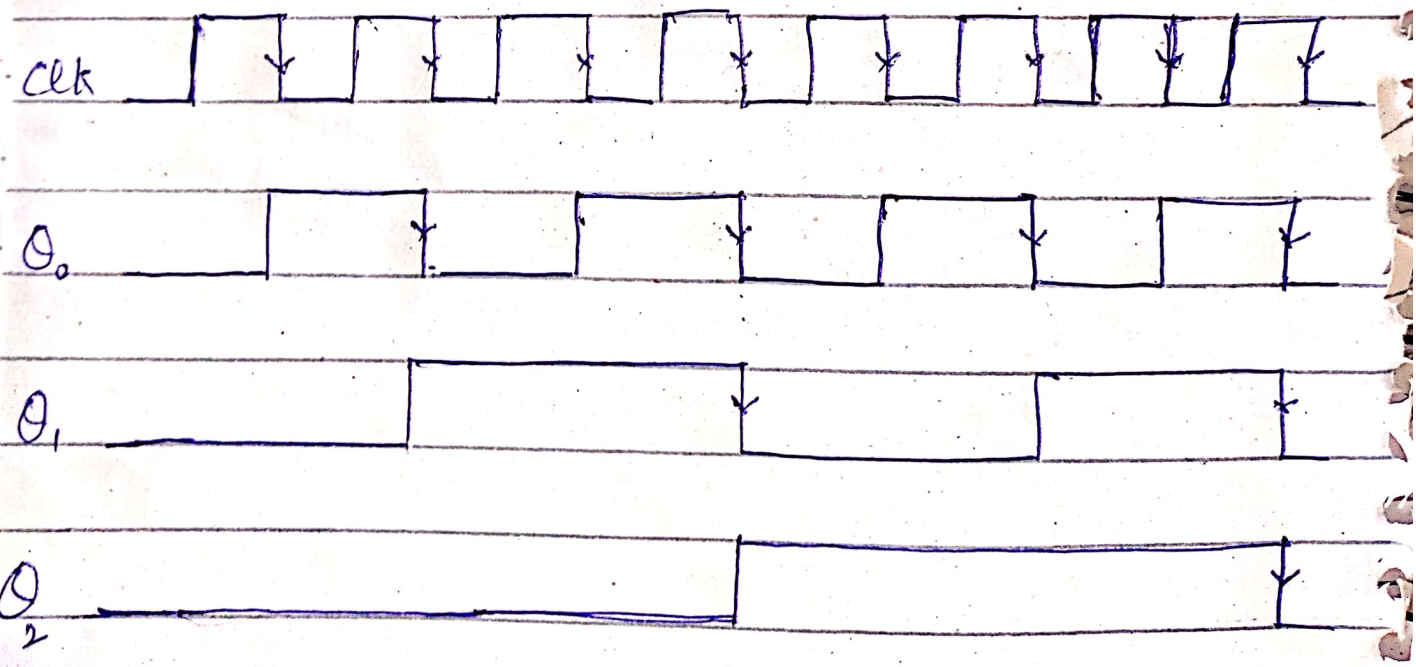
• Diagram:



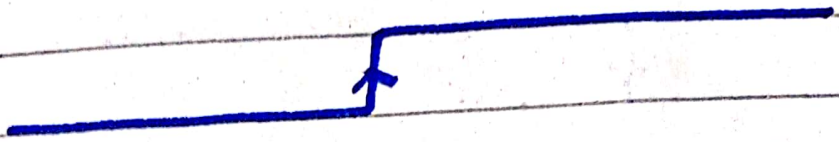
- Table:

Clock	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Timing Diagram



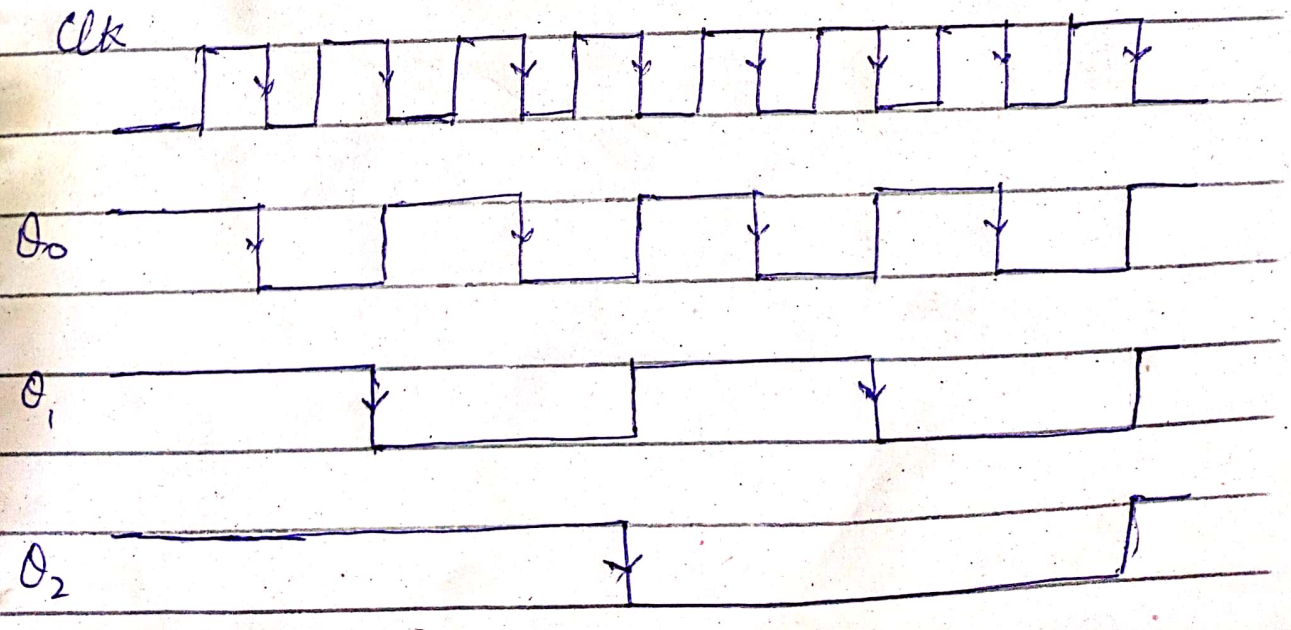
Q_2 :



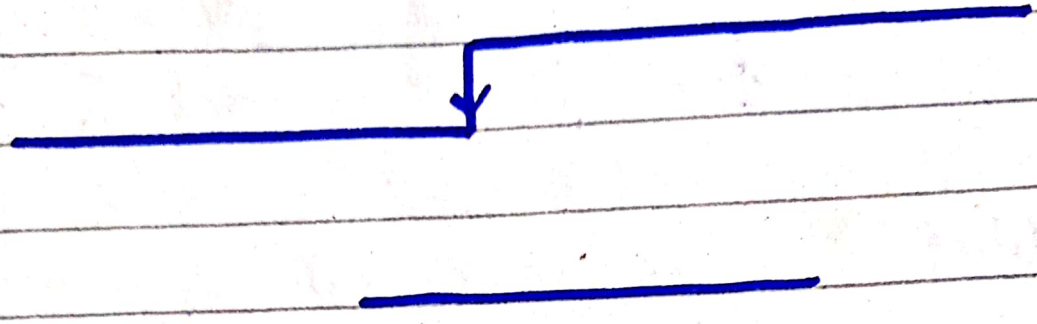
- Table:

Clock	Q_2	Q_1	Q_0
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

Timing Diagram



• Q_2:



• 3-bit Synchronous up Counter by using FF.

1- Find number of flip flops required and select type of FF.
2- Write truth table for selected FF and excitation table.
3- Construct Stable table with corresponding excitation table.
4- Build K-map for each FF input & simplify K-map and obtain minimized expression.
5- Draw circuits.

• Truth Table:

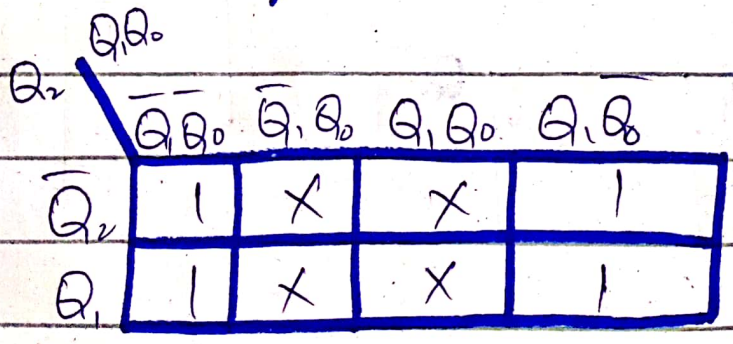
is on next Page

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_0K_0	J_1K_1	J_2K_2
0	0	0	0	0	1	1X	0X	0X
0	0	1	0	1	0	X1	1X	0X
0	1	0	0	1	1	1X	X0	0X
0	1	1	1	0	0	X1	X1	1X
1	0	0	1	0	1	1X	0X	X0
1	0	1	1	1	0	X1	1X	X0
1	1	0	1	1	1	1X	X0	X0
1	1	1	0	0	0	X1	X1	X1

- **Excitation Table for JK, FE**

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

- **K-map:**



$J_0 = 1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
Q_2	X	1	1	X
$\bar{Q}_2$	X	1	1	X

$K_0 = 1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
Q_2	X	X	1	0
$\bar{Q}_2$	X	X	1	0

$K_1 = Q_0$

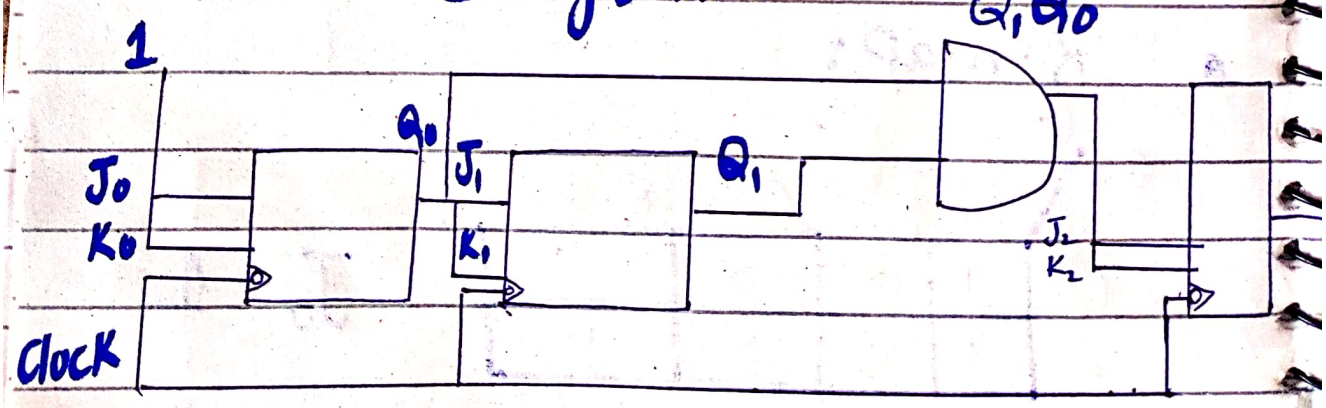
	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
Q_2	0	1	X	X
$\bar{Q}_2$	0	1	X	X

$J_1 = Q_0$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
Q_2	0	0	1	0
$\bar{Q}_2$	X	X	X	X

$J_2 = Q_1Q_0$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
Q_2	X	X	X	X
$\bar{Q}_2$	0	0	1	0

$K_2 = Q_1Q_0$

- **Circuit Diagram:**